Midterm Problems: For Fun

- I claim that every card with a B on the front has a 5 on the back. Of the following cards, how many and which do you have to flip over to tell whether my rule is true for these cards?

B	K	5	3

1

First-Order Logic & Higher-Order Logic, Knowledge-Based Agents

Material from Dr. Marie desJardin, Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer

2

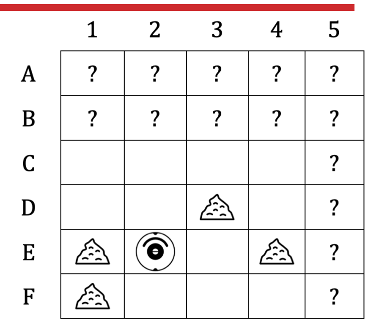
Bookkeeping

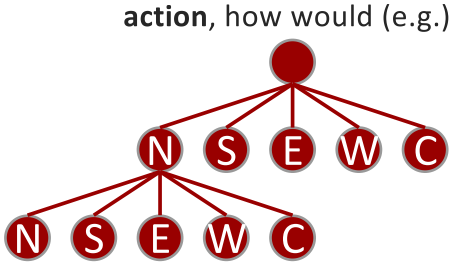
- HW4 out on 10/31 🎃🪔, due 11/20
- **No HW5**—effort should be going into project at this point
- Designs will be graded this week—read the comments!
  - Some people will be asked to turn in a second version
  - Please do this ASAP
- Today's class:
  - Midterm review (quickly)
  - Review/finalize propositional logic
  - Converting to CNF
  - First-order logic
  - Knowledge-based agents

3

Midterm Problems: Agents and Search

- Possible actions are N, S, E, W, and Clean
  - So branching factor is…
- For search algorithms, the "and why" mattered
  - Did you convey an understanding of the algorithm and how it would work in this world?
  - **Given that the robot must interleave exploration and action**, how would (e.g.) DFS work?





UCS: **complete** with path costs of 0?

4

Midterm Problems: Local Search

- State space
  - What is a single state (like the starting state)?
  - And another one you can get to from there?
  - What then is the set of all possible states?
  -
- Core operation is…
- Evaluation function should measure **for any given state** whether that state is "good"—e.g., how many boxes are filled? Is everything in a box?
- Most local search algorithms work; all have local minima

5

Midterm Problems: Constraint Satisfaction

		B	A	S	E
+		B	A	L	L
	G	A	M	E	S

- What are the random variables?
  - That is, what are you trying to assign values to?
- What's the domain? That is, what values can they take?
- What are the **explicit** constraints? What do we know about values?

6

Midterm Problems: Constraint Satisfaction

- What's the Minimum Remaining Values heuristic?
  - Start by assigning values to variables that have the fewest remaining available values

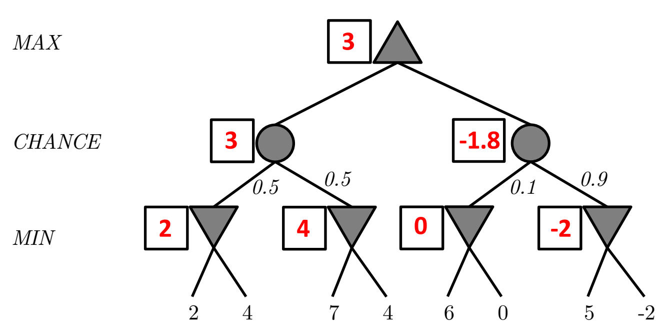
		B	A	S	E
+		B	A	L	L
	G	A	M	E	S

- How would you apply it here?
  - Take one of B, A, S, E, L, G, M; which has fewest available values?
-

7

Midterm Problems: Expectiminimax

- Utility values are backed up the tree, multiplied by the appropriate chance value
- 15 (alpha-beta pruning) was **not graded or counted**—chance nodes mean you need bounds on the evaluation function to prune anything (this is in the book)



8

Midterm Problems: Nash Equilibria

- Some people got confused with Pareto Optimality (not the same thing!)
- No player benefits by **unilaterally** changing strategy while others stay fixed
- Nash equilibria are +1/-1. Let's go case by case:

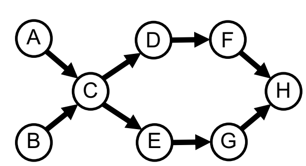
	Straight	Swerve
Straight	-10, -10	+1, -1
Swerve	-1, +1	0, 0

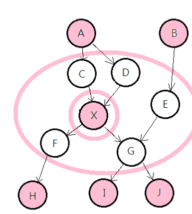
ht

9

Midterm Problems: Bayesian Belief Networks

- Capture relationships among nodes: **directed** edges mean a node is influenced by another node
  - Eating questionable food increases the odds of having an upset stomach
- A node is conditionally independent of other nodes in the network given its parents, children, and children's parents (its Markov blanket)
  - Is D $\perp\!\!\!\perp$ E | A, B? No—they share a parent
  - Is A $\perp\!\!\!\perp$ C | D? No—C is a child of A
  - Is A $\perp\!\!\!\perp$ H | C? Yes—C "deactivates" the connection





10

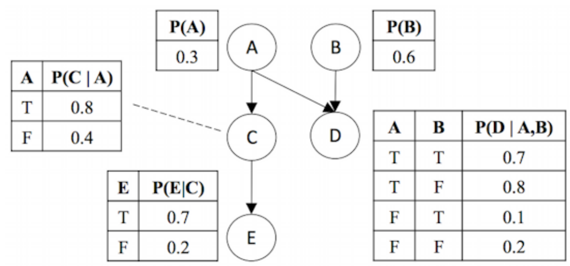
Midterm Problems: Bayesian Belief Networks

P(A , D | B) =

P(D|A , B) P(A|B) =

P(D|A , B) P(A) =

(since A and B are independent)



0.8*0.3 = 0.24

- Please consider 21 and try to work through it, **given independences**
  - Start with P(A|C) = P(C|A)P(A) / P(C)

11

First-Order Logic

Chapter 8

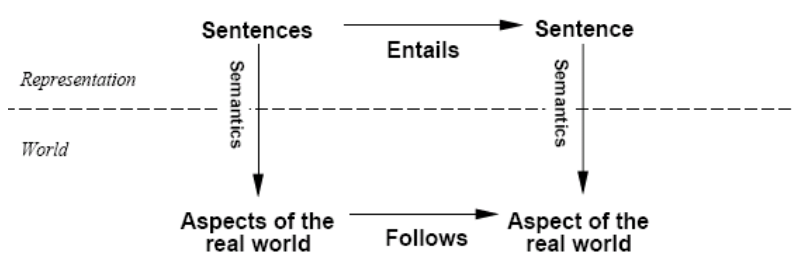
12

Review

- Definitions:
  - Syntax, Semantics, Sentences, Propositions, Entails, Follows, Derives, Inference, Sound, Complete, Model, Satisfiable, Valid (or Tautology), etc.
- Syntactic Transformations:
  - E.g., $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$
- Truth Tables
  - Negation, Conjunction, Disjunction, Implication, Equivalence (Biconditional)
  - Inference by Model Enumeration

13

Review: Schematic perspective



If KB is true in the real world,
then any sentence α ***entailed*** *by KB*
is also true in the real world.

For example: If I tell you (1) Sue is Mary's sister, and (2) Sue is Amy's mother, then it necessarily follows in the world that Mary is Amy's aunt, even though I told you nothing at all about aunts. This sort of reasoning pattern is what we hope to capture.

14

Review: Logic

- If a problem domain can be represented formally, then a decision maker can use **logical reasoning** to make rational decisions
- Many types of logic
  - Propositional Logic (Boolean logic)
  - First-Order Logic (aka first-order predicate calculus)
  - Non-Monotonic Logic
  - Markov Logic
- A logic includes:
  - syntax: what is a correctly-formed sentence?
  - semantics: what is the meaning of a sentence?
  - Inference procedure (reasoning, entailment): what sentence logically follows given knowledge?

15

Review: Propositional Logic

- A **symbol** in Propositional Logic (PL) is a symbolic variable whose value must be either True or False, and which stands for a natural language statement that could be either true or false
  - A = “Smith has chest pain”
  - B = “Smith is depressed”
  - C = “It is raining”

16

Review: Semantics

- An **interpretation** is a complete True / False assignment to all propositional symbols
  - Example symbols: P means "It is hot", Q means "It is humid", R means "It is raining"
  - There are 8 interpretations (TTT, ..., FFF)
- The semantics (meaning) of a sentence is the set of interpretations in which the sentence evaluates to True
- Example: the semantics of the sentence P ∨ Q is the set of 6 interpretations:
  - P=True, Q=True, R=True or False
  - P=True, Q=False, R=True or False
  - P=False, Q=True, R=True or False
- **A model of a set of sentences is an interpretation in which all the sentences are true**

17

Review: Knowledge Base (KB)

- A knowledge base, KB, is a set of sentences
  - Example KB:
  - HaveLecture ⇔ (TodayIsTuesday ∨ TodayIsThursday)
  - ¬HaveLecture
- It is equivalent to a single long sentence: the conjunction of all sentences
  - (HaveLecture ⇔ (TodayIsTuesday ∨ TodayIsThursday)) ∧ ¬HaveLecture
- **A model of a KB is an interpretation in which all sentences in KB are true**

18

Review: Entailment

- **Entailment** is the relation of a sentence β ***logically following*** from other sentences α (e.g., KB): **α ⊨ β**
- α ⊨ β if and only if, in every interpretation in which α is true, β is also true; i.e., whenever α is true, so is β
- Deduction theorem: **α ⊨ β if and only if α ⇒ β is valid** (always true)
- Proof by contradiction (refutation, *reductio ad absurdum*): **α ⊨ β if and only if α ∧ ¬β is unsatisfiable**
- There are 2^n interpretations to check, if KB has *n* symbols

19

Review: Entailment vs. Inference

- If your knowledge base KB entails p, then all interpretations that evaluate KB to True also evaluate p to True
  - (interpretation = assignment of ‘true’ or ‘false’ values to variables)
  - KB ⊨ p
- Inference is a procedure for deriving a new sentence q from KB following some algorithm
  - KB ⊢ q
  - Inference is **sound** if it derives only sentences that are entailed by the KB
  - Inference is **complete** if anything entailed by the KB can also be inferred from the KB

20

So—how do we stay correct?

Is this inference correct?

How do you know?
How can you tell?

How can we **make correct** inferences?
How can we **avoid incorrect** inferences?

"Einstein Simplified: Cartoons on Science" by Sydney Harris, 1992, Rutgers University Press

21

So—how do we stay correct?

- All men are people;
- Half of all people are women;
- Therefore, half of all men are women.

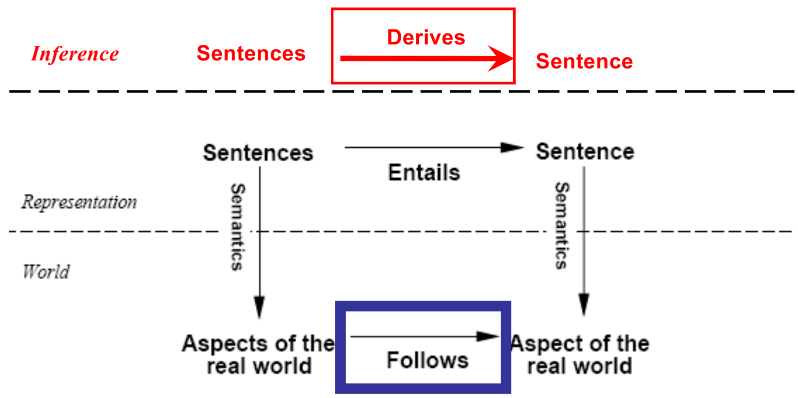
Is this inference correct?

How do you know?
How can you tell?

- Penguins are black and white;
- Some old TV shows are black and white;
- Therefore, some penguins are old TV shows.

22

Schematic perspective



If KB is true in the real world,

then any sentence α derived from KB

by a sound inference procedure

is also true in the real world.

Conjunctive Normal Form (CNF)

We'd like to prove: KB ⊨ α
(This is equivalent to KB ∧ ¬ α is unsatisfiable.)

We first rewrite $KB \wedge \neg\alpha$ into **conjunctive normal form (CNF).**

A "conjunction of disjunctions"

(A ∨ ¬B) ∧ (B ∨ ¬C ∨ ¬D)

Clause — Clause

literals

- Any KB can be converted into CNF.

Review: Equivalence & Implication

- Equivalence is a conjoined double implication
  - **(X ⇔ Y) = [(X ⇒ Y) ∧ (Y ⇒ X)]**
- Implication is (NOT antecedent OR consequent)
  - **(X ⇒ Y) = (¬X ∨ Y)**
  - How do we know this is true?
  - We can always use a truth table:

p	q	¬p	p ⇒ q	¬p V q
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

25

Review: de Morgan's rules

- How to bring ¬ inside parentheses
  - (1) Negate everything inside the parentheses
  - (2) Change operators to "the other operator"
- ¬(X ∧ Y ∧ … ∧ Z) = (¬X ∨ ¬Y ∨ … ∨ ¬Z)
- ¬(X ∨ Y ∨ … ∨ Z) = (¬X ∧ ¬Y ∧ … ∧ ¬Z)

26

Review: Boolean Distributive Laws

- Both of these laws are valid:
- AND distributes over OR
  - $X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$
  - $(W \vee X) \wedge (Y \vee Z) = (W \wedge Y) \vee (X \wedge Y) \vee (W \wedge Z) \vee (X \wedge Z)$
- OR distributes over AND
  - $X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$
  - $(W \wedge X) \vee (Y \wedge Z) = (W \vee Y) \wedge (X \vee Y) \wedge (W \vee Z) \wedge (X \vee Z)$

27

Conjunctive Normal Form (CNF)

1. Replace all $\Leftrightarrow$ using iff/biconditional elimination
   - $\alpha \Leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$
2. Replace all $\Rightarrow$ using implication elimination
   - $\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$
3. Move all negations inward using
   - double-negation elimination
     - $\neg(\neg\alpha) \equiv \alpha$
   - de Morgan's rule
     - $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$
     - $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$
4. Apply distributivity of $\vee$ over $\wedge$
   - $\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ + 1 more

28

Example Conversion of Sentence into CNF

- $A \Leftrightarrow (B \vee C)$ — starting sentence
- $(A \Rightarrow (B \vee C)) \wedge ((B \vee C) \Rightarrow A)$ — iff/biconditional elimination
- $(\neg A \vee B \vee C) \wedge (\neg(B \vee C) \vee A)$ — implication elimination
- $(\neg A \vee B \vee C) \wedge ((\neg B \wedge \neg C) \vee A)$ — move negations inward
- $(\neg A \vee B \vee C) \wedge (\neg B \vee A) \wedge (\neg C \vee A)$ — distribute $\vee$ over $\wedge$

called a "**clause**"

29

Example: Conversion to CNF

Example: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$ by replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

   $= (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$ by replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$ and simplify.

   $= (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and simplify.

   $$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta), \neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$$

   $= (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributive law ($\wedge$ over $\vee$) and simplify.

   $= (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

30

Example: Conversion to CNF

- Example: B1,1 ⇔ (P1,2 ∨ P2,1)
- From the previous slide we had:
  - = (¬B1,1 ∨ P1,2 ∨ P2,1) ∧ (¬P1,2 ∨ B1,1) ∧ (¬P2,1 ∨ B1,1)

5. KB is the conjunction of all of its sentences (all are true),
   - so write each clause (disjunct) as a sentence in KB:

- KB = …
  (¬B1,1 ∨ P1,2 ∨ P2,1)
  (¬P1,2 ∨ B1,1)
  (¬P2,1 ∨ B1,1)
  …

Often, Won't Write "∧"
(we know it is there)

- Can do this in Propositional Logic, but often we want to use First Order Logic

31

First-Order Logic

- First-order logic (FOL) models the world in terms of
  - **Objects**, which are things with individual identities
  - **Properties** of objects that distinguish them from other objects
  - **Relations** that hold among sets of objects
  - **Functions**, which are a subset of relations where there is only one "value" for any given "input"
- Examples:
  - Objects: students, lectures, companies, cars …
  - Relations: brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, …
  - Properties: blue, oval, even, large, …
  - Functions: father-of, best-friend, second-half, one-more-than …

32

FOL Contains

- **Constant symbols**, which represent individuals in the world
  - Mary
  - 3
  - Green
- **Function symbols**, which map individuals to individuals
  - father-of(Mary) = John
  - color-of(Sky) = Blue
- **Predicate symbols**, which map individuals to truth values
  - greater(5,3)
  - green(Grass)
  - color(Grass, Green)

33

FOL Contains

- **Variable symbols**
  - E.g., x, y, foo
- **Connectives**
  - Same as in PL: not ($\neg$), and ($\wedge$), or ($\vee$), implies ($\Rightarrow$), if and only if (biconditional $\leftrightarrow$)
- **Quantifiers**
  - Universal $\forall$x or (Ax)
  - Existential $\exists$x or (Ex)

34

Sentences: Terms and Atoms

- A **term** (denoting a real-world individual) is:
  - A constant symbol: *John*, or
  - A variable symbol: *x*, or
  - An n-place function of n terms
    - x and $f(x_1, \ldots, x_n)$ are terms, where each x_i is a term
    - *is-a(John, Professor)*
  - A term with no variables is a **ground term**.
- An **atomic sentence** is an n-place predicate of n terms
  - Has a truth value (*t* or *f*)

35

First-Order Logic (FOL

- Propositional logic assumes the world contains facts.
- First-order logic (like natural language) assumes the world contains
  - **Objects:** people, houses, numbers, colors, baseball games, wars, …
  - **Functions**: father of, best friend, one more than, plus, …
    - Function arguments are objects; function returns an object
  - Objects generally correspond to English NOUNS
  - **Predicates/Relations/Properties:** red, round, prime, brother of, bigger than, part of, between…
    - Predicate arguments are objects; predicate returns a truth value
  - Predicates generally correspond to English VERBS
    - **First argument is generally the subject, the second the object**
    - Hit(Bill, Ball) usually means "Bill hit the ball."
    - Likes(Bill, IceCream) usually means "Bill likes IceCream."
    - Verb(Noun1, Noun2) usually means "Noun1 verb noun2."

36

Syntax of FOL: Atomic Sentences

- Atomic sentences in logic state facts that are true or false.
- Properties and *n*-ary relations do just that:
  - LargerThan(2, 3) is false.
  - Married(Father(Richard), Mother(John)) could be true or false.
- Note: Functions refer to objects, do not state facts, and form no sentence:
  - Brother(Pete) refers to John (his brother) and is neither true nor false.
  - Plus(2, 3) refers to the number 5 and is neither true nor false.
- BrotherOf(Pete, Brother(Pete)) is True.

↑ Binary relation is a truth value.

↑ Function refers to John, an object in the world, i.e., John is Pete's brother.

37

Syntax of FOL: Variables

- **Variables** range over objects in the world.
- A **variable** is like a **term** because it represents an object.
- A **variable** may be used wherever a **term** may be used.
  - **Variables** may be arguments to functions and predicates.
- (A **term with NO variables** is called a **ground term**.)
- (A **variable not bound by a quantifier** is called **free**.

38

Syntax of FOL: Basic syntax elements are symbols

- Constant Symbols (correspond to English nouns)
  - Stand for objects in the world. E.g., KingJohn, 2, France, ...
- Predicate Symbols (correspond to English verbs)
  - Stand for relations (maps a tuple of objects to a truth-value)
    - E.g., Brother(Richard, John), greater_than(3,2), ...
- Function Symbols (correspond to English nouns)
  - Stand for functions (maps a tuple of objects to an object)
    - E.g., Sqrt(3), LeftLegOf(John), ...
- Model (world) = set of domain objects, relations, functions
- Interpretation maps symbols onto the model (world)
  - Very many interpretations are possible for each KB and world!
  - Job of the KB is to rule out models inconsistent with our knowledge.

39

Syntax of FOL: Basic elements

Constants	KingJohn, 2, UMBC,...	
Predicates	BrotherOf, >,...	(return true or false)
Functions	Sqrt, LeftLegOf,...	(return some object)
Variables	x, y, a, b,...	
Quantifiers	$\forall$, $\exists$	
Connectives	$\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ (standard)	
Equality	= (but causes difficulties....)	

40

Sentences: Terms and Atoms

- A **complex sentence** is formed from atomic sentences connected by the same logical connectives as in propositional logic:
  $\neg$P, P$\vee$Q, P$\wedge$Q, P$\Rightarrow$Q, P$\leftrightarrow$Q where P and Q are sentences

has-a(x, Bachelors) $\wedge$ *is-a(x, human)*

does NOT SAY everyone with a bachelors' is human

what **does** it say?

has-a(John, Bachelors) $\wedge$ *is-a(John, human)*

has-a(Mary, Bachelors)

41

Quantifiers

- **Universal quantification**
  - ∀x P(x) means that P holds for **all** values of *x* in its domain
  - States universal truths
  - E.g.: ∀*x dolphin(x)* $\Rightarrow$ *mammal(x)*
- **Existential quantification**
  - ∃x P(x) means that P holds for **some** value of x in the domain associated with that variable
  - Makes a statement about some object without naming it
  - E.g., ∃*x mammal(x)* $\wedge$ *lays-eggs(x)*

42

Sentences: Quantification

- **Quantified sentences** adds quantifiers ∀ and ∃

  $\forall x$ *has-a(x, Bachelors)* $\Rightarrow$ *is-a(x, human)*

  $\exists x$ *has-a(x, Bachelors)*

  $\forall x\ \exists y$ *Loves(x, y)*

Everyone who has a bachelors' is human.

There exists some who has a bachelors'.

Everybody loves somebody.

43

Sentences: Well-Formedness

- A **well-formed formula (wff)** is a sentence containing no "free" variables. That is, all variables are "bound" by universal or existential quantifiers.

- $(\forall x)P(x,y)$ has x bound as a universally quantified variable, but y is free: It is NOT wff

44

Quantifiers: Uses of ∀

- Universal quantifiers **often** used with “implies” to form “rules”:

  $(\forall x)$ student(x) $\Rightarrow$ smart(x)

  “All students are smart”

- Universal quantification **rarely*** used to make blanket statements about every individual in the world:

  $(\forall x)$student(x)$\wedge$smart(x)

  “Everyone in the world is a student and is smart”

*Deliberately, anyway

45

Quantifiers: Uses of ∃

- Existential quantifiers are **usually** used with “and” to specify a list of properties about an individual:
  - $(\exists x)$ student(x) $\wedge$ smart(x)
  - “There is a student who is smart”
- A common mistake is to represent this English sentence as the FOL sentence:
  - $(\exists x)$ student(x) $\Rightarrow$ smart(x)
  - But what happens when there is a person who is not a student?

46

Translation with Quantifiers

- Universal statements typically use **implications**
  - All S(x) is P(x):
    - $\forall x(S(x) \Rightarrow P(x))$
  - No S(x) is P(x):
    - $\forall x(S(x) \Rightarrow \neg P(x))$
- Existential statements typically use **conjunctions**
  - Some S(x) is P(x):
    - $\exists x (S(x) \wedge P(x))$
  - Some S(x) is not P(x):
    - $\exists x (S(x) \wedge \neg P(x))$

47

Quantifier Scope

- Switching the order of universal quantifiers does not change the meaning:
  - $(\forall x)(\forall y)P(x,y) \leftrightarrow (\forall y)(\forall x) P(x,y)$
- Similarly, you can switch the order of existential quantifiers:
  - $(\exists x)(\exists y)P(x,y) \leftrightarrow (\exists y)(\exists x) P(x,y)$
- Switching the order of universals and existentials does change meaning:
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x,y)
  - Someone is liked by everyone: $(\exists y)(\forall x)$ likes(x,y)

48

Connections between All and Exists

- We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws:
  - $(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$
  - $\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$
  - $(\forall x)\ P(x) \leftrightarrow \neg(\exists x)\ \neg P(x)$
  - $(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$

49

Quantified Inference Rules

- Universal instantiation
  - $\forall x\ P(x)\ \therefore\ P(A)$
- Universal generalization
  - $P(A) \wedge P(B) \ldots\ \therefore\ \forall x\ P(x)$
- Existential instantiation
  - $\exists x\ P(x)\ \therefore P(F)$ **← skolem constant F**
- Existential generalization
  - $P(A)\ \therefore\ \exists x\ P(x)$

50

Universal Instantiation (a.k.a. Universal Elimination)

- If (∀x) P(x) is true, then P(C) is true, where C is any constant in the domain of x
- Example:
  - (∀x) eats(Ziggy, x) ⇒ eats(Ziggy, IceCream)
- The variable symbol can be replaced by any ground term, i.e., any constant symbol or function symbol applied to ground terms only

51

Existential Instantiation (a.k.a. Existential Elimination)

- Variable is replaced by a **brand-new constant**
  - I.e., not occurring in the KB
- From (∃x) P(x) infer P(c)
  - Example:
    - (∃x) eats(Ziggy, x) ⇒ eats(Ziggy, Stuff)
  - "Skolemization" – create a new term that instantiates existence
- *Stuff* is a **skolem constant**
- Easier than manipulating the existential quantifier

52

Existential Generalization (a.k.a. Existential Introduction)

- If P(c) is true, then $(\exists x)$ P(x) is inferred.
- Example
  - eats(Ziggy, IceCream) $\Rightarrow$ $(\exists x)$ eats(Ziggy, x)
- All instances of the given constant symbol are replaced by the new variable symbol
- Note that the variable symbol cannot already exist anywhere in the expression

53

Translating English to FOL

- **Every gardener likes the sun.**
  - $\forall x$ gardener(x) $\Rightarrow$ likes(x,Sun)
- **You can fool some of the people all of the time.**
  - $\exists x\ \forall t$ person(x) $\wedge$time(t) $\Rightarrow$ can-fool(x,t)
- **You can fool all of the people some of the time.**
  - $\forall x\ \exists t$ (person(x) $\Rightarrow$ time(t) $\wedge$can-fool(x,t))
  - $\forall x$ (person(x) $\Rightarrow$ $\exists t$ (time(t) $\wedge$can-fool(x,t))

  Equivalent
- **All purple mushrooms are poisonous.**
  - $\forall x$ (mushroom(x) $\wedge$ purple(x)) $\Rightarrow$ poisonous(x)

54

Translating English to FOL

- **No purple mushroom is poisonous.**
  - $\neg\exists x\ purple(x) \land mushroom(x) \land poisonous(x)$
  - $\forall x\ (mushroom(x) \land purple(x)) \Rightarrow \neg poisonous(x)$

  Equivalent
- **There are exactly two purple mushrooms.**
  - $\exists x\ \exists y\ mushroom(x) \land purple(x) \land mushroom(y) \land purple(y) \wedge \neg(x=y) \land \forall z\ (mushroom(z) \land purple(z)) \Rightarrow ((x=z) \lor (y=z))$
- **Mary is not tall.**
  - $\neg tall(Mary)$
- **X is above Y iff X is on directly on top of Y or there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.**
  - $\forall x\ \forall y\ above(x,y) \leftrightarrow (on(x,y) \lor \exists z\ (on(x,z) \land above(z,y)))$

55

Semantics of FOL

- **Domain M:** the set of all objects in the world (of interest)
- **Interpretation I:** includes
  - Assign each constant to an object in M
  - Define each function of n arguments as a mapping M^n => M
  - Define each predicate of n arguments as a mapping M^n => {T, F}
  - Therefore, every ground predicate with any instantiation will have a truth value
  - In general there is an infinite number of interpretations because |M| is infinite
- **Define logical connectives:** ~, ^, v, =>, <=> as in PL
- **Define semantics of (∀x) and (∃x)**
  - (∀x) P(x) is true iff P(x) is true under all interpretations
  - (∃x) P(x) is true iff P(x) is true under some interpretation

56

Terminology

- **Model**: an interpretation of a set of sentences such that every sentence is True
- A sentence is
  - **Satisfiable** if it is true under some interpretation
  - **Valid** if it is true under all possible interpretations
  - **Inconsistent** if there does not exist any interpretation under which the sentence is true
- **Logical consequence**: $S \models X$ if all models of S are also models of X

57

Axioms, Definitions and Theorems

- **Axioms** are facts and rules that attempt to capture all of the (important) facts and concepts about a domain; axioms can be used to prove **theorems**
  - Mathematicians don't want any unnecessary (dependent) axioms –ones that can be derived from other axioms
  - Dependent axioms can make reasoning faster, however
  - Choosing a good set of axioms for a domain is a kind of design problem
- A definition of a predicate is of the form "p(X) ↔ …" and can be decomposed into two parts
  - **Necessary** description: "p(x) ⇒ …"
  - **Sufficient** description "… ⇒ p(x)"
  - Some concepts don't have complete definitions (e.g., person(x))

58

Necessary and Sufficient

- p is **necessary** for q
  - ¬p ⇒ ¬q (“no p, no q!”)
- p is **sufficient** for q
  - p ⇒ q (“p is all we need to know!”)
- Note that ¬p ⇒ ¬q is equivalent to q ⇒ p
- So if p is necessary and sufficient for q, then p iff q.

59

More on Definitions

- Examples: define father(x, y) by parent(x, y) and male(x)
  - parent(x, y) is a **necessary (but not sufficient**) description of father(x, y)
    - father(x, y) ⇒ parent(x, y)
  - parent(x, y) ^ male(x) ^ age(x, 35) is a **sufficient (but not necessary**) description of father(x, y):
    - father(x, y) ⟵ parent(x, y) ^ male(x) ^ age(x, 35)
  - parent(x, y) ^ male(x) is a **necessary and sufficient** description of father(x, y)
    - parent(x, y) ^ male(x) ⟷ father(x, y)

60

Converting FOL to CNF

- Eliminate biconditionals and implications
- Move ¬ inwards
- **Standardize variables apart** by renaming them: each quantifier should use a different variable
- **Skolemize:** each existential variable is replaced by a Skolem constant or Skolem function of the enclosing universally quantified variables.
  - For instance, ∃x Rich(x) becomes Rich(G1) where G1 is a new Skolem constant
  - "Everyone has a heart" [∀x Person(x) ⇒ ∃y Heart(y)∧Has(x,y)] becomes ∀x Person(x) ⇒ Heart(H(x)) ∧ Has(x, H(x)), where H is a new symbol (Skolem function)
- **Drop universal quantifiers**
  - For instance, ∀ x Person(x) becomes Person(x).
- Distribute ∧ over ∨

61

Summary: First Order Logic (FOL)

- Uses the same logical symbols as Propositional Logic (PL)
- Adds: variables, quantification, predicates and functions
  - Names of terms: **constants**, **variables**, **predicates**, **functions**
- Existential and universal quantifiers can be used to create rules
- Need to be able to translate English to and from FOL
- Some extensions…

62

Higher-Order Logic

- FOL only allows to quantify over variables, and variables can only range over objects.
- HOL allows us to quantify over relations
- Example: (quantify over functions)
  - "two functions are equal iff they produce the same value for all arguments"
  - $\forall f\ \forall g\ (f = g) \leftrightarrow (\forall x\ f(x) = g(x))$
- Example: (quantify over predicates)
  - $\forall r$ transitive(r) $\Rightarrow (\forall xyz)\ r(x,y) \wedge r(y,z) \Rightarrow r(x,z))$
- More expressive, but undecidable.

63

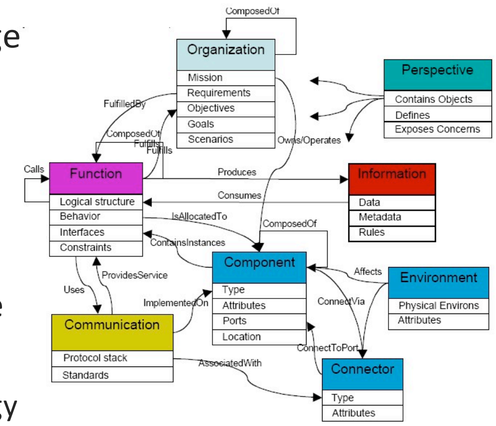
Expressing Uniqueness

- Sometimes we want to say that there is a single, unique object that satisfies a certain condition
- "There exists a unique x such that king(x) is true"
  - $\exists x\ king(x) \wedge \forall y\ (king(y) \Rightarrow x{=}y)$
  - $\exists x\ king(x) \wedge \neg\exists y\ (king(y) \wedge x \neq y)$
- Iota operator: "ι x P(x)" means "the unique x such that p(x) is true"
  - "The unique ruler of Freedonia is dead"
  - dead(ι x ruler(freedonia,x))

64

Knowledge bases/ontologies

- Ontology: the study of what there is—an inventory of what exists
- **An** ontology: a hierarchical categorization system for things in the world
- A formally represented corpus of knowledge
  - Defined by some *grammar*
  - Incorporates *rules* (implicitly or explicitly)
  - Not divided into tables: more like a *graph*
  - Often hierarchical
- Usually incorporate background knowledge (not purely domain-specific)
  - Although many are **in** a domain, such as biology



65

65

Ontological knowledge

- Assertions
  - (isa ...

d:

R)

  - (implies (isa ?X Cup) (isa ?X Container))
- Combine them to draw conclusions:
  - (isa ?WHAT Mug-17) → ??????

We'll do more on ontologies when we get to knowledge representation.

This is a formal representation space than can underpin questions like "What type of thing is a mug?" or "Who is Dr. M's brother?

66

Notational differences

- Different symbols for and, or, not, implies, ...
  - ∀ ∃ ⇒ ⇔ ∧ ∨ ¬ • ⊃
  - p v (q ^ r)
  - p + (q * r)
  - etc
- Prolog
  - cat(X) :- furry(X), meows (X), has(X, claws)
- Lispy notations

```
(forall ?x (implies (and (furry ?x)
                         (meows ?x)
                         (has ?x claws))
                    (cat ?x)))
```

67

Exercise: FOL translation

1. Everything is bitter or sweet.
2. Either everything is bitter or everything is sweet.
3. There is somebody who is loved by everyone.
4. Nobody is loved by no one.
5. If someone is noisy, everybody is annoyed
6. Frogs are green.
7. Frogs are not green.
8. No frog is green.
9. Some frogs are not green.
10. A mechanic likes Bob.
11. A mechanic likes herself.
12. Every mechanic likes Bob.
13. Some mechanic likes every nurse.
14. There is a mechanic who is liked by every nurse.

Exercises: disi.unitn.it/~bernardi/Courses/LSNL/Slides/fl1.pdf

68

Exercise: FOL translation

1. $\forall x\, (bitter(x) \vee sweet(x))$
2. $\forall x\, (bitter(x)) \vee \forall x\, (sweet(x))$
3. $\exists x\, \forall y\, (loves(y,x))$
4. $\neg\exists x\, \neg\exists y\, (loves(y,x))$
5. $\exists x\, (noisy(x)) \Rightarrow \forall y(annoyed(y))$
6. $\forall x\, (frog(x) \Rightarrow green(x))$
7. $\forall x\, (frog(x) \Rightarrow \neg green(x))$
8. $\neg\exists x\, (frog(x) \wedge green(x))$
9. $\exists x\, (frog(x) \wedge \neg green(x))$
10. $\exists x\, (mech.(x) \wedge likes(x, Bob))$
11. $\exists x\, (mech.(x) \wedge likes(x, x))$
12. $\forall x\, (mech.(x) \Rightarrow likes(x, Bob))$
13. $\exists x\, \forall y\, (mech(x) \wedge nurse(y) \Rightarrow likes(x, y))$
14. $\exists x(mech(x) \wedge \forall y(nurse(y) \Rightarrow likes(y, x))$

Exercises: disi.unitn.it/~bernardi/Courses/LSNL/Slides/fl1.pdf

69