15-780: Graduate AI
Lecture 3. FOL proofs; SAT

Geoff Gordon (this lecture)
Tuomas Sandholm
TAs Byron Boots, Sam Ganzfried

Admin

HW1

- *Out today*
- *Due Tue, Feb. 3 (two weeks)*
  - *hand in hardcopy at beginning of class*
- *Covers propositional and FOL*
- *Don't leave it to the last minute!*

Collaboration policy

- *OK to discuss general strategies*
- *What you hand in must be your own work*
  - *written with no access to notes from joint meetings, websites, etc.*
- *You must acknowledge all significant discussions, relevant websites, etc., on your HW*

Late policy

- *You have 3 late days in total to split across all HWs*
  - *these account for conference travel, holidays, illness, or any other reasons*
- *After late days, 75% for next day, 50% for next, 0% thereafter (but still must turn in)*
- *Day = 24 hrs, HWs due at 10:30AM*

Office hours

- *Office hours start this week (see website for times)*
- *But, I have a conflict this week due to admissions; let me know by email if there is demand, and if so I can reschedule*

Matlab tutorial

- *Thu 1/22, 4–5PM, Wean Hall 5409*

Review

In propositional logic

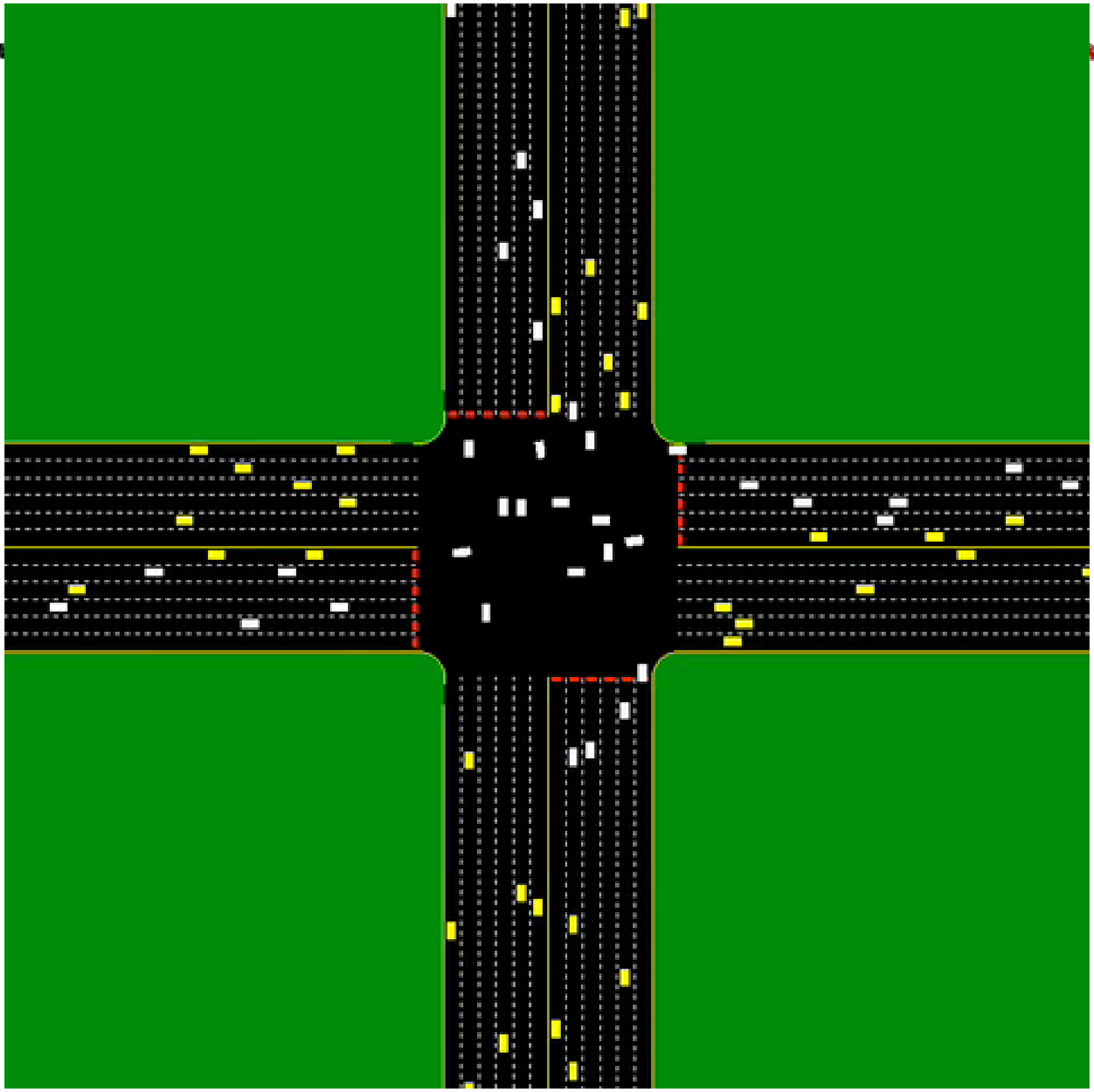
- *Compositional semantics, structural induction*
- *Proof trees, proof by contradiction*
- *Inference rules (e.g., resolution)*
- *Soundness, completeness*
- *Horn clauses*
- *Nonmonotonic logic*

In FOL

- *Compositional semantics*
  - *objects, functions, predicates*
  - *terms, atoms, literals, sentences*
  - *quantifiers, free/bound variables*
  - *models, interpretations*
- *Generalized de Morgan's law*
- *Skolemization, CNF*

Project Ideas

Traffic insanity



Sensor planning

- *Plan a path for this robot so that it gets a good view of an object as fast as possible*

Mini-robots

- *Do something cool w/ Lego Mindstorms*
  - *plan footstep placements*
  - *plan how to grip objects*

Poker

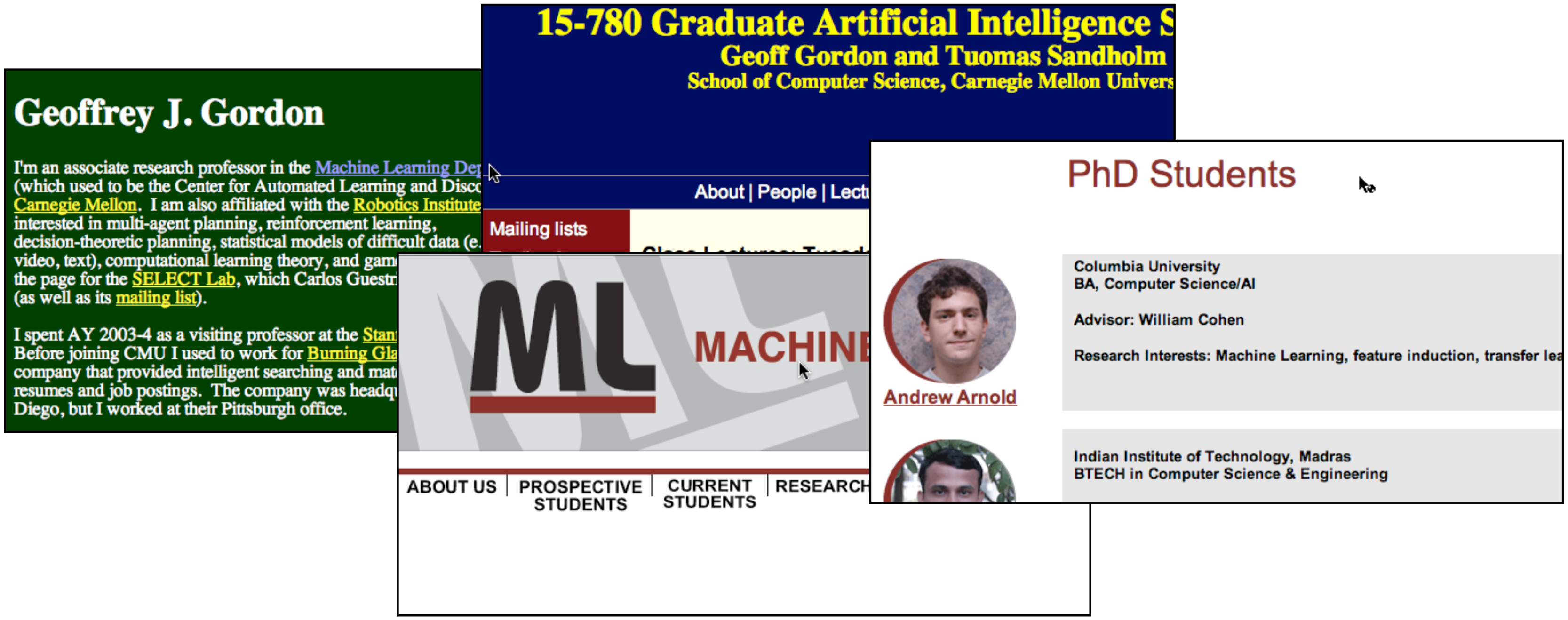
Poker

- *Minimax strategy for heads-up poker = solving linear program*
- *1-card hands, 13-card deck: 52 vars, <u>instantaneous</u>*
- *RI Hold'Em: ~1,000,000 vars*
  - *2 weeks / 30GB (exact sol, CPLEX)*
  - *40 min / 1.5GB (approx sol)*
- *TX Hold'Em: ??? (up to 10^{17} vars or so)*

Poker

- *Learning by repeated play*
  - *we'll discuss learning algorithms later*
- *Possibly state-of-the-art for 2 players*
- *We don't know another feasible approach for 3 or more players*
- *Project: pick a poker domain, compare several learning algorithms and/or other solution methods*

Understand the web



- *Write a probabilistic knowledge base describing a portion of the web*
- *Learn parameters of the model*

Proofs in FOL

FOL is special

- *Despite being much more powerful than propositional logic, there is still a **sound** and **complete** inference procedure for FOL*
- *Almost any significant extension breaks this property*
- *This is why FOL is popular: very powerful language with a sound & complete inference procedure*

Proofs

- *Proofs by contradiction work as before:*
  - *add ¬S to KB*
  - *put in CNF*
  - *run resolution*
  - *if we get an empty clause, we've proven S by contradiction*
- *But, CNF and resolution have changed*

Generalizing resolution

- *Propositional:* $(\neg a \lor b) \land a \models b$
- *FOL:*

  $(\neg man(x) \lor mortal(x)) \land man(Socrates)$

  $\models (\neg man(Socrates) \lor mortal(Socrates))$
  $\land\ man(Socrates)$

  $\models mortal(Socrates)$
- *Difference: had to substitute* $x \rightarrow Socrates$

Universal instantiation

- *What we just did is UI:*

  $(\neg man(x) \lor mortal(x))$

  $\vDash (\neg man(Socrates) \lor mortal(Socrates))$

- *Works for x → any ground term*

  $(\neg man(uncle(student(Socrates))) \lor$
  $mortal(uncle(student(Socrates))))$

- *For proofs, need a good way to find useful instantiations*

Substitution lists

- *List of variable → value pairs*
- *Values may contain variables (leaving flexibility about final instantiation)*
- *But, no LHS may be contained in any RHS*
  - *i.e., applying substitution twice is the same as doing it once*
- *E.g., $x \rightarrow$ Socrates, $y \rightarrow$ LCA(Socrates, z)*

LCA = last common advisor

Unification

- *Two FOL terms* ***unify*** *with each other if there is a substitution list that makes them syntactically identical*
- *man(x), man(Socrates) unify using the substitution* $x \rightarrow$ *Socrates*
- *Importance: purely syntactic criterion for identifying useful substitutions*

Unification examples

- *loves(x, x), loves(John, y) unify using* $x \to John, y \to John$
- *loves(x, x), loves(John, Mary) can't unify*
- *loves(uncle(x), y), loves(z, aunt(z)):*

Unification examples

- $loves(x, x)$, $loves(John, y)$ *unify using* $x \rightarrow John, y \rightarrow John$
- $loves(x, x)$, $loves(John, Mary)$ *can't unify*
- $loves(uncle(x), y)$, $loves(z, aunt(z))$:
  - $z \rightarrow uncle(x), y \rightarrow aunt(uncle(x))$
  - $loves(uncle(x), aunt(uncle(x)))$

Quiz

- *Can we unify*

  knows(John, x)   knows(x, Mary)

- *What about*

  knows(John, x)   knows(y, Mary)

Quiz

- *Can we unify*

  $$knows(John, x) \quad knows(x, Mary)$$

  No!

- *What about*

  $$knows(John, x) \quad knows(y, Mary)$$

  $$x \rightarrow Mary, y \rightarrow John$$

Standardize apart

- *But knows(x, Mary) is logically equivalent to knows(y, Mary)!*
- *Moral: standardize apart before unifying*

Most general unifier

- *May be many substitutions that unify two formulas*
- *MGU is unique (up to renaming)*
- *Simple, moderately fast algorithm for finding MGU (see RN); more complex, linear-time algorithm*

Linear unification. MS Paterson, MN Wegman. Proceedings of the eighth annual ACM symposium on Theory of Computing, 1976.

First-order resolution

- *Given clauses* $(a \vee b \vee c)$, $(\neg c' \vee d \vee e)$, *and a substitution list V unifying c and c'*
- *Conclude* $(a \vee b \vee d \vee e) : V$

Example

$$rains \wedge outside(x) \Rightarrow wet(x)$$

$$wet(x) \Rightarrow rusty(x) \vee rustproof(x)$$

$$robot(x) \Rightarrow \neg rustproof(x)$$

$$rains$$

$$guidebot(Robby)$$

$$guidebot(x) \Rightarrow robot(x) \wedge outside(x)$$

$rains \wedge outside(x) \Rightarrow wet(x)$

6 $\neg rains \vee \neg outside(x) \vee wet(x)$

$wet(x) \Rightarrow rusty(x) \vee rustproof(x)$

11 $\neg wet(y) \vee rusty(y) \vee rustproof(y)$

$robot(x) \Rightarrow \neg rustproof(x)$

4 $\neg robot(z) \vee \neg rustproof(z)$

8 $rains$

1 $guidebot(Robby)$

$guidebot(x) \Rightarrow robot(x) \wedge outside(x)$

2 $\neg guidebot(a) \vee robot(a)$

3 $\neg guidebot(b) \vee outside(b)$

$\neg(\exists x . rusty(x))$

14 $\neg rusty(c)$

1,2 $\models$ 5 $robot(Robby)$

1,3 $\models$ 7 $outside(Robby)$

4,5 $\models$ 10 $\neg rustproof(Robby)$

6,7 $\models$ 9 $\neg rains \vee wet(R)$

8,9 $\models$ 12 $wet(Robby)$

10,11 $\models$ 13 $\neg wet(Robby) \vee rusty(R)$

12,13 $\models$ 15 $rusty(R)$

14,15 $\models$ F

First-order factoring

- *When removing redundant literals, we have the option of unifying them first*
- *Given clause* $(a \vee b \vee c)$*, substitution* V
- *If* $a : V$ *and* $b : V$ *are the same*
- *Then we can conclude* $(a \vee c) : V$

Completeness

- *First-order resolution (together with first-order factoring) is sound and complete for FOL*
- *Famous theorem*

Completeness

Proof strategy

- *We'll show FOL completeness by reducing to propositional completeness*
- *To prove S, put KB $\wedge \neg$S in clause form*
- *Turn FOL KB into propositional KBs*
  - *in general, infinitely many*
- *Check each one in order*
- *If any one is unsatisfiable, we will have our proof*

Propositionalization

- *Given a FOL KB in clause form*
- *And a set of terms U (for **universe**)*
- *We can **propositionalize** KB under U by substituting elements of U for free variables in all combinations*

Propositionalization example

- $(\neg man(x) \vee mortal(x))$
- $man(Socrates)$
- $favorite\_drink(Socrates) = hemlock$
- $drinks(x, favorite\_drink(x))$

- $U = \{Socrates, hemlock, Fred\}$

Propositionalization example

- $(\neg man(Socrates) \vee mortal(Socrates))$
  $(\neg man(Fred) \vee mortal(Fred))$
  $(\neg man(hemlock) \vee mortal(hemlock))$
- $drinks(Socrates, favorite\_drink(Socrates))$
  $drinks(hemlock, favorite\_drink(hemlock))$
  $drinks(Fred, favorite\_drink(Fred))$
- $man(Socrates) \wedge$
  $favorite\_drink(Socrates) = hemlock$

Choosing a universe

- *To check a FOL KB, propositionalize it using some universe U*
- *Which universe?*

Herbrand Universe

Jacques Herbrand
1908–1931

- ***Herbrand universe*** *H of formula S:*
  - *start with all objects mentioned in S*
  - *or synthetic object X if none mentioned*
  - *apply all functions mentioned in S to all combinations of objects in H, add to H*
  - *repeat*

Herbrand Universe

- *E.g., loves(uncle(John), Mary) yields*

  H = {John, Mary, uncle(John), uncle(Mary), uncle(uncle(John)), uncle(uncle(Mary)), … }

Herbrand's theorem

- *If a FOL KB in clause form is unsatisfiable*
- *And H is its Herbrand universe*
- *Then the propositionalized KB is unsatisfiable for some* ***finite*** $U \subseteq H$

Significance

- *This is one half of the equivalence we want: unsatisfiable FOL KB ⇒ ∃ finite U. unsatisfiable propositional KB*

Example

- $(\neg man(x) \lor mortal(x)) \land man(uncle(Socrates)) \land \neg mortal(x)$
- $H = \{S, u(S), u(u(S)), \ldots\}$
- *If* $U = \{u(S)\}$, *PKB* $=$

  $(\neg man(u(S)) \lor mortal(u(S))) \land man(u(S)) \land \neg mortal(u(S))$
- *Resolving twice yields F*

Converse of Herbrand

- *A. J. Robinson proved "lifting lemma"*
- *Write PKB for a propositionalization of KB (under some universe)*
- *Any resolution proof in PKB corresponds to a resolution proof in KB*
- *… and, if PKB is unsatisfiable, there is a proof of F (by prop. completeness); so, lifting it shows KB unsatisfiable*

Example

- $(\neg man(u(S)) \vee mortal(u(S))) \wedge man(u(S)) \wedge \neg mortal(u(S))$
- *We resolved on man(u(S)) yielding mortal(u(S))*
- *Lifted, resolve* $\neg man(x)$ *w/* $man(u(S))$*, binding* $x \rightarrow u(S)$

Proofs w/ Herbrand & Robinson

- *So, FOL KB is unsatisfiable* ***if and only if*** *there is a subset of its Herbrand universe making PKB unsatisfiable*
- *I.e., if we have a way to find proofs in propositional logic, we have a way to find them in FOL*

Proofs w/ Herbrand & Robinson

- *To prove S, put KB ∧ ¬S in CNF: KB'*
- *Build subsets of Herbrand universe in increasing order of size:* $U_1, U_2, \ldots$
- *Propositionalize KB' w/* U_i*, look for proof*
- *If* U_i *unsatisfiable, use lifting to get a contradiction in KB'*
- *If* U_i *satisfiable, move on to* U_{i+1}

How long will this take?

- *If S is not entailed, we will never find a contradiction*
- *In this case, if H infinite, we'll never stop*
- *So, entailment is* ***semidecidable***
  - *equivalently, entailed statements are* ***recursively enumerable***

Variation

- *Restrict semantics so we only need to check one finite propositional KB*
- ***Unique names****: objects with different names are different (John ≠ Mary)*
- ***Domain closure****: objects without names given in KB don't exist*
- *Restrictions also make entailment, validity feasible*

Who? What? Where?

Wh-questions

- *We've shown how to answer a question like "is Socrates mortal?"*
- *What if we have a question whose answer is not just yes/no, like "who killed JR?" or "where is my robot?"*
- *Simplest approach: prove* $\exists x.\ killed(x, JR)$*, hope the proof is constructive*

55

Answer literals

- *Simple approach doesn't always work*
- *Instead of ¬S(x), add (¬S(x) ∨ answer(x))*
- *If there's a contradiction, we can eliminate ¬S(x) by resolution and unification, leaving answer(x)* ***with x bound*** *to a value that causes a contradiction*

56

Example

$kills(Jack, Cat) \vee kills(Curiosity, Cat)$

$\neg kills(Jack, x)$

[redacted]

1 kills(Jack, Cat) ∨ kills(Curiosity, Cat)

2 ¬kills(Jack, x)

3 ¬kills(x, Cat)

1,3 x → Jack ⊨ 4 kills(Curiosity, Cat)

3,4 x → Curiosity ⊨ F

---

5 ¬kills(x, Cat) ∨ answer(x)

1,5 x → Curiosity ⊨ 6 kills(Jack, cat) ∨ answer(Curiosity)

2,6 x → cat ⊨ answer(Curiosity)

58

FOL Extensions

Equality

- ***Paramodulation*** *is sound and complete for FOL+equality (see RN)*
- *Or, resolution +* ***axiom schema***

Second order logic

- *SOL adds quantification over predicates*
- *E.g., principle of mathematical induction:*
  - $\forall P.\ P(0) \land (\forall x.\ P(x) \Rightarrow P(S(x)))$
    $\Rightarrow \forall x.\ P(x)$
- *There is no sound and complete inference procedure for SOL (Gödel's famous incompleteness theorem)*

Others

- *Temporal logics ("P(x) will be true at some time in the future")*
- *Modal logics ("John believes P(x)")*
- *Nonmonotonic FOL*
- *First-class functions (lambda operator, application)*
- …

Using FOL

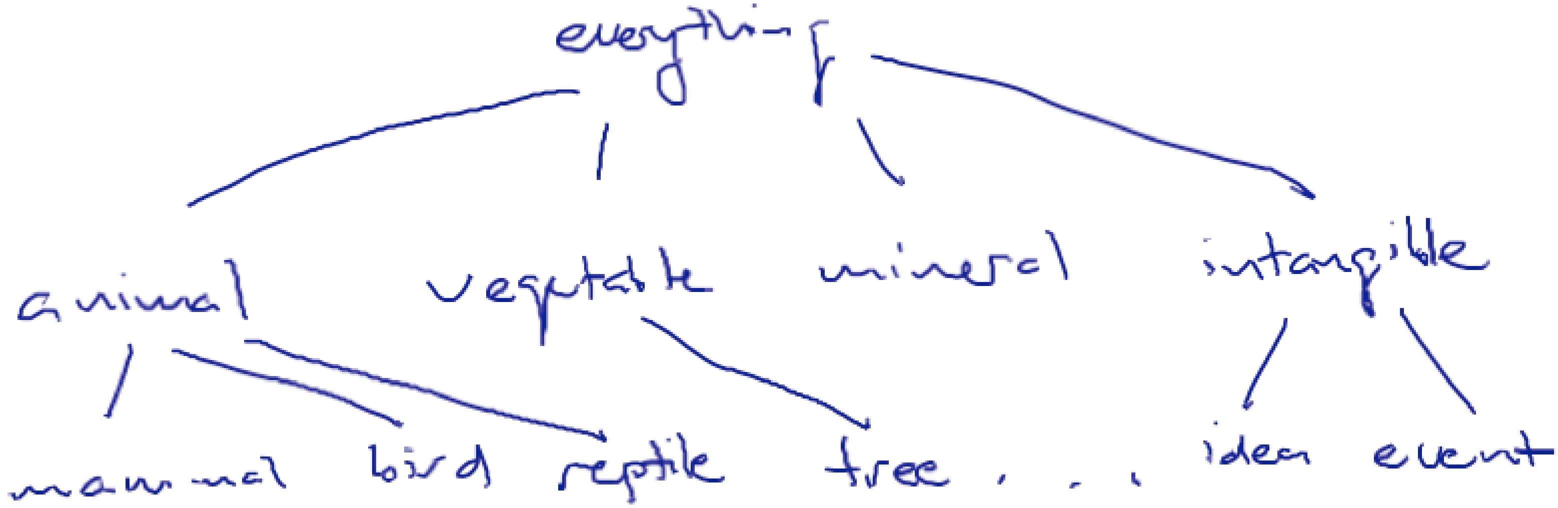
Knowledge engineering

- *Identify relevant objects, functions, and predicates*
- *Encode general background knowledge about domain (reusable)*
- *Encode specific problem instance*
- *Pose queries (is P(x) true? Find x such that P(x))*

Common themes

- *RN identifies many common idioms and problems for knowledge engineering*
- *Hierarchies, fluents, knowledge, belief, …*
- *We'll look at a couple*

Taxonomies



- *isa(Mammal, Animal)*
- *disjoint(Animal, Vegetable)*
- *partition({Animal, Vegetable, Mineral, Intangible}, Everything)*

Inheritance

- *Transitive:* $isa(x, y) \wedge isa(y, z) \Rightarrow isa(x, z)$
- *Attach properties anywhere in hierarchy*
  - $isa(Pigeon, Bird)$
  - $isa(x, Bird) \Rightarrow flies(x)$
  - $isa(x, Pigeon) \Rightarrow gray(x)$
- *So, isa(Tweety, Pigeon) tells us Tweety is gray and flies*

Physical composition

- *partOf(Wean4625, WeanHall)*
- *partOf(water37, water)*
- *Note distinction between **mass** and **count** nouns: any partOf a mass noun is also an example of that same mass noun*

Fluents

- *Fluent = property that changes over time*
  - *at(Robot, Wean4623, 11AM)*
- *Actions change fluents*
- *Fluents chain together to form possible worlds*
- $at(x, p, t) \wedge adj(p, q) \Rightarrow poss(go(x, p, q), t) \wedge at(x, q, result(go(x, p, q), t))$

Frame problem

- *Suppose we execute an unrelated action (e.g., talk(Professor, FOL))*
- *Robot shouldn't move:*
  - *if at(Robot, Wean4623, t), want at(Robot, Wean4623, result(talk(Professor, FOL)))*
- *But we can't prove it using tools described so far!*

Frame problem

- *The **frame problem** is that it's a pain to list all of the things that don't change when we execute an action*
- *Naive solution: **frame axioms***
  - *for each fluent, list actions that can't change fluent*
  - *KB size: O(AF) for A actions, F fluents*

Frame problem

- *Better solution:* ***successor-state*** *axioms*
- *For each fluent, list actions that* ***can*** *change it (typically fewer): if go(x, p, q) is possible,*

  $at(x, q, result(a, t)) \Leftrightarrow$
  $a = go(x, p, q) \lor (at(x, q, t) \land a \neq go(x, q, z))$
- *Size O(AE+F) if each action has E effects*

Sadly, also necessary…

- *Debug knowledge base*
  - *Severe bug: logical contradictions*
  - *Less severe: undesired conclusions*
  - *Least severe: missing conclusions*
- *First 2: trace back chain of reasoning until reason for failure is revealed*
- *Last: trace desired proof, find what's missing*