CSC421 Intro to Artificial Intelligence

UNIT 16:
Knowledge Engineering

Outline

- Ontological Engineering
- Categories & Objects
- Physical Composition
- Measurements
- Substances & Objects

Representing everything - the big picture

- Anything
  - AbstractObjects
    - Sets
      - Categories
    - Numbers
    - RepresentationalObjects
      - Sentences
      - Measurements
        - Times
        - Weights
  - Generalized Events
    - SEE NEXT SLIDE

The big picture

Ontologies

- Upper ontology
- FOL challenges
  - exceptions
  - uncertainty
- Do all ontologies converge on a general purpose ontology
- After centuries of discussion possibly
- General ontology
  - Applicable to any specific domain with addition of domain specific axioms
  - Different areas of ontologies must be “unified”
- CYC 6000 concepts with 60,000 facts

Categories & Objects

- Much reasoning in terms of categories
- Predictions about objects once classified
- Green, mottled skin, large size, ovoid shape
  - Infer it is a watermellon (category)
  - Useful for fruit salad (predictions about specific object based on category characteristics)
- Two choices
  - Predicate: Basketball(b)
  - Object (reify): Basketballs
- The two main ideas:
  - Subset(Basketballs, Balls)
  - Member(b, Basketballs)

More about categories & objects

- Inheritance
  - All foods are edible
  - Fruit is a subclass of Food and Apples is a subclass of Fruit then we know all apples are edible
- Taxonomy
  - Used for centuries – examples
    - Biology
    - Dewey Decimal System
    - Musical genres

FOL and categories/objects

- An object is a member of a category
  - This ball is a baskball
- A category is a subclass of another category
  - Basketballs are a type of ball.
- All members of of a category have some property
  - All basketballs are round
- Members of a category can be recognized by some properties
  - Orange and round balls are basketballs
- A category as a whole has some properties

FOL and categories/objects

- An object is a member of a category
  - Bbg $\in$ Basketballs
- A category is a subclass of another category
  - Basketballs $\subset$ Balls
- All balls are round
  - x $\in$ Basketballs => Round(x)
- Orange and round balls are basketballs
  - Orange(x) $\wedge$ Round(x) $\wedge$ x $\in$ Balls => x $\in$ BBalls

Relations between categories

- Disjoint = no members in common
  - Disjoint(animals, vegetables)
- Exhaustive decomposition
  - Exhaustive Decomposition({American, Canadians, Mexicans}, NorthAmericans)
- Disjoint exhaustive decomposition or partition
  - Partition({Males, Females}, Animals)
- Exercise (without reading book)
  - Try to define in FOL

Physical Composition

- PartOf relation
- Composite objects – structural relations
- Composite objects without structure
  - The apples in this bag weigh two pounds
- A set of apples is a math concept doesn't have weight
- Bunch of apples (apples as parts not elements)

Measurement

- Height, mass, cost etc
  - Values for properies = measures
- “Same” length has different “names”
  - 1.5 inches 3.81 cm
- Length(L1) = Inches(1.5) = Centimeters (3.81)
- Centimeters(2.54 x d) = Inches(d)
- Qualitative physics
- Nominal, ordinal, categorical

Natural Kinds

- Tomatoes tend to be red
- Only true for "typical" instances
- member(x, Typical(Tomatoes)) => Red(x) ^ Round(x)

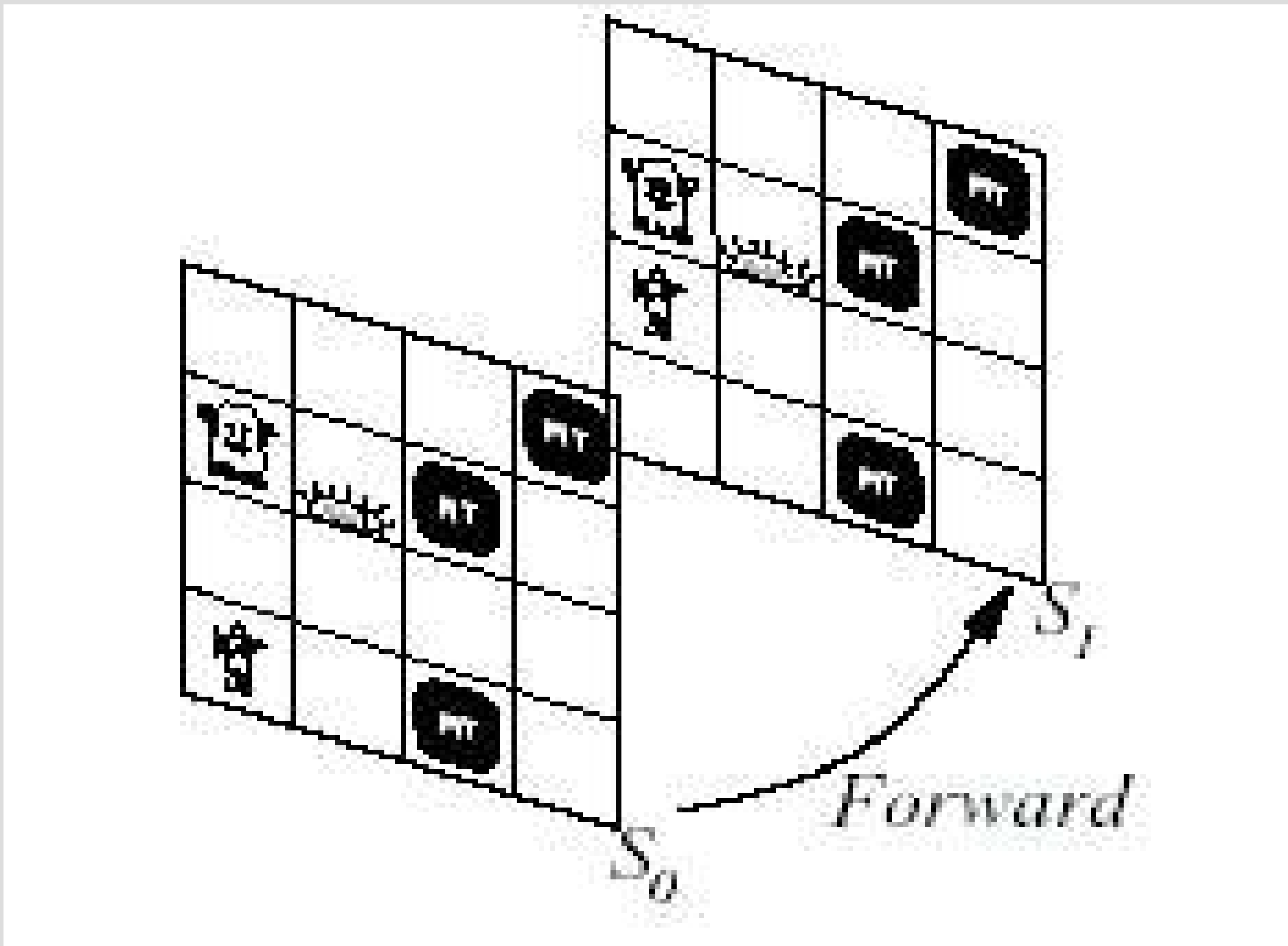
Substances and Objects

- "Stuff"
  - Water, butter
  - Count nouns
    - Horses, theorems, basketballs
  - Mass nouns
    - Energy, butter, water
- Intrinsic properties (only => mass noun)
  - Salted, yellow, tasty
- Extrinsic properties (any => count noun)
  - Weight, length, shape
- Stuff vs Thing
- All physical objects belong to both categories

Keeping track of change

- Facts hold in situations, rather than eternally
  - e.g. Holding(Gold, Now) rather than just Holding(Gold)
- Situation calculus is one way to represent change in FOL
  - Add a situation argument to every non-eternal predicate (for example Now is a situation)
- Situations are connected by the Result function
  - Result(a,s) is the situation that results from doing a in s

Situation Calculus



Describing axioms I

- “Effect” axiom – describe changes due to action
  - $\forall$s AtGold(s) => Holding(Gold, Result(Grab, s))
- “Frame” axioms – describe non-changes due to action
  - $\forall$s HaveArrow(s) => HaveArrow(Result(Grab, s))
- Frame problem : find elegant way to handle non-change (identified in 1969)
  - Representation : avoid frame axioms
  - Inference: avoid repeated “copy-overs” to keep track of state

Describing actions II

- Succesor-state axioms solve the representational frame problem
  - Each axiom is about a predicate
    - [P true afterwards] iff [an action made P true or P true already and no action made P false]
  - For holding the gold:
  - $\forall$a,s Holding(Gold, Result(a,s)) <=>

    [a=Grab $\wedge$ AtGold(s)] $\vee$ [Holding (Gold) $\wedge$ a $\neq$ Release]
- R.Reiter 1991 UofT

Making Plans

- Initial condition in KB:
  - At(Agent, [1,1], S0)
  - At(Gold, [1,2], S0)
- Query: Ask(KB, ∃ s Holding(Gold, s))
  - i.e in what situation will I be holding Gold
  - Answer {s/Result(Grab, Result(Forward, So))}
- This assumes that the agent is interested in plans starting at S0 and that S0 is the only situation in KB

Making Plans: a better way

- Plans = actions sequences
- PlanResult(p,s) is the resulting of executing p in s
- Then the query Ask(KB, ∃ p Holding(Gold, PlanResult(p, S0))
- Answer: {p / [Forward, Grab]}
- Define PlanResult in terms of Result

Definition of PlanResult

- In terms of Result
  - $\forall s \quad PlanResult([], s) = s$
  - $\forall a, p, s \quad PlanResult([a|p], s) = PlanResult(p, Result(a,s))$
- Planning systems are special purpose reasoners designed to do this type of inference more efficiently than a general purpose reasoner