



More Knowledge Representation

CS156

Chris Pollett

Nov 12, 2014

Outline

- Categories and Objects
- Events
- Reasoning Systems

Introduction

- On Monday, we started talking about frameworks, which we called **ontologies**, for representing facts about the world so that they can be used by logic-based agents.
- We mentioned four attempts to come up with one-size-fits-all ontologies which work for all domains.
- Today, we are going to look at common approaches for particular domains. We will not be as exhaustive as the book, but hopefully give some flavor of this area.
- Then we will discuss some reasoning systems based on these.

Categories and Objects

- In this approach, we try to group objects in the world into categories. For example, the category of objects which are Basketballs.
- These are often represented in first-order logic as predicates. Any b that satisfies $Basketball(b)$ would be deemed a basketball.
- Another approach is to imagine that there is an object which collects together all objects which are basketballs (this approach is called **reification**). Call this object Basketballs. We can then ask for an object b whether $b \in Basketballs$.
- We write $Basketballs \subset Balls$ to indicate Basketballs is a **subcategory/subclass/subset** of Balls.
- A subcategory **inherits** all the properties of its parent category. i.e., what is true of the parent class is true of the child.
- Categories can be organized into **taxonomies** by using the subclass relation.
- The kinds of facts that we can state in taxonomies are: (a) an object is a member of a category, (b) a category is a subclass of another, (c) all members of a category have some property (For example, $x \in BasketBall \Rightarrow Spherical(x)$), (d) Members of a category can be recognized by some property (For example, $Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5'' \wedge x \in Ball \Rightarrow x \in BasketBall$), and a category as a whole has some properties (for example, $Dogs \in DomesticatedSpecies$).
- We will often decompose categories into subcategories. If every object in the original belongs to one of the subcategories we say we have an **exhaustive decomposition**. If, in addition, the subcategories are **disjoint** (share no objects), we call the decomposition a **partition**.

Physical Composition

- We now consider how to represent knowledge about physical objects
- One common way to group objects is to say that they are part of some other. We will use the **PartOf** relation for this purpose. For example, PartOf(Bucharest, Romania), PartOf(Romania, EasternEurope).
- **Composite Objects** can be represented by asserting its parts exists and their relationships:

$$\begin{aligned} Biped(a) \Leftrightarrow & \exists l_1, l_2, b \text{Leg}(l_1) \wedge \text{Leg}(l_2) \wedge \text{Body}(b) \wedge \\ & \text{PartOf}(l_1, a) \wedge \text{PartOf}(l_2, a) \wedge \text{PartOf}(b, a) \wedge \\ & \text{Attached}(l_1, b) \wedge \text{Attached}(l_2, b) \wedge \\ & l_1 \neq l_2 \wedge (\forall l_3 \wedge \text{Leg}(l_3) \wedge \text{PartOf}(l_3, a) \Rightarrow (l_3 = l_1 \vee l_3 = l_2)) \end{aligned}$$
- A PartPartition of an object can be defined in a similar way: we assert the existence of various components to the object, we are assert members of these components are disjoint and that the collection of all component members is the whole.
- It is also sometimes useful to define composite objects with definite parts but no structure. We can use the BunchOf relation for this. For example, BunchOf({Apple₁, Apple₂, Apple₃}).
- BunchOf can be defined in terms of PartOf using:

$$\forall x, x \in s \Rightarrow \text{PartOf}(x, \text{BunchOf}(s)) \text{ and } \forall y, [\forall x, x \in s \Rightarrow \text{PartOf}(x, y)] \Rightarrow \text{PartOf}(\text{BunchOf}(s), y) .$$
- The second statement ensures BunchOf is the smallest relation satisfying the first property. This is what's called **logical minimization**.

Measurements

- In both scientific and commonsense theories of the world, objects have height, mass, cost, etc.
- To model these, we imagine the world has abstract measure objects such as *length*.
- Often the same length, has different values in different units. For example 1in = 2.54cm. We represent units using **units functions** such as *Inches*(d).
- Conversion between units can be done using conversion functions:
 $Inches(d) = Centimeters(2.54 \times d)$.
- We can now use measures to describe objects with expressions like:
 $Diameter(Basketball_3) = Inches(9.5)$.
 $ListPrice(Basketball_2) = \(19) .
 $d \in Days \Rightarrow Duration(d) = Hours(24)$.
- Often what one uses to reason about measure is an ordering $>$ rather than the specific values.

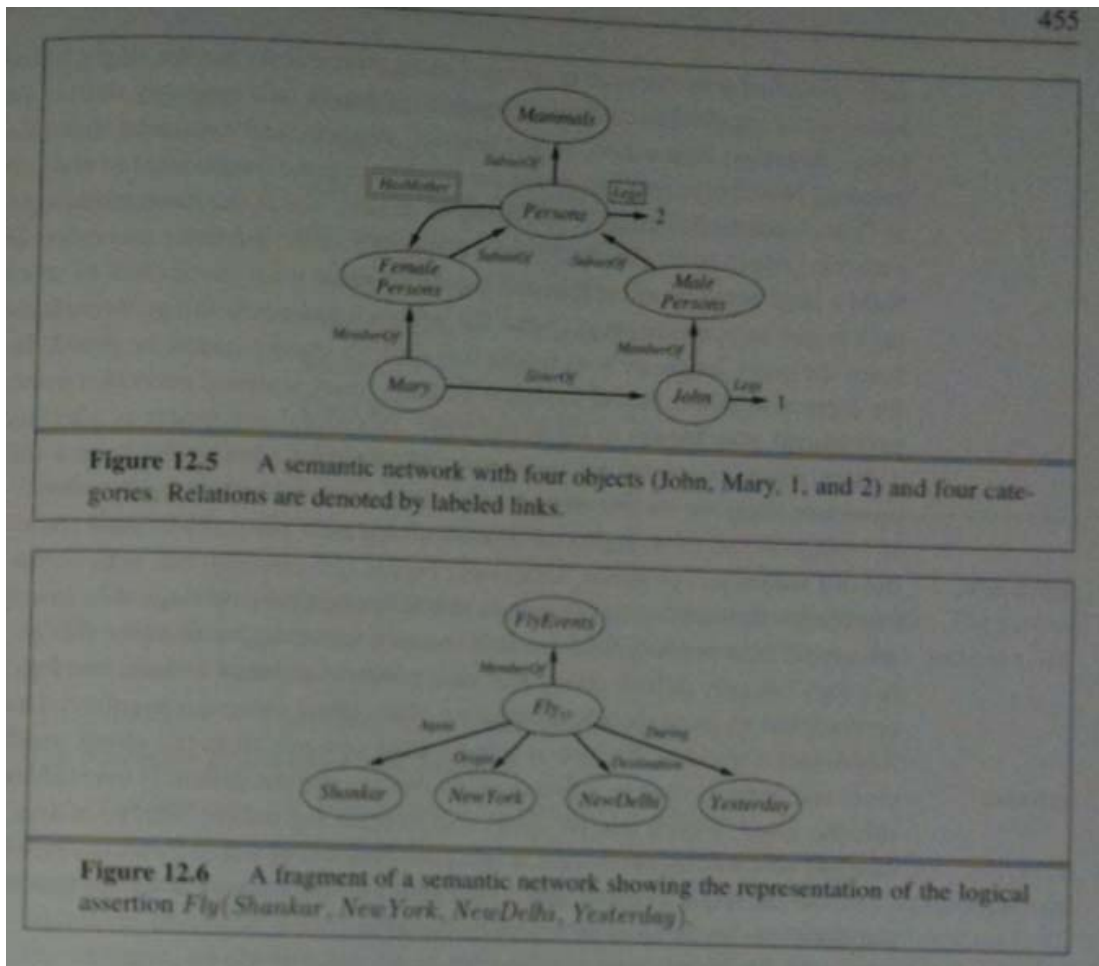
Events

- We would like to be able to express things like the state of the world at a given time and also facts about things that were true during the time it took for an action to take place (unlike say in PDDL or things like the situation calculus which can't say anything except before and after events.)
- To do this we will briefly an **event calculus**.
- Events for us will be instances of the events category.
- As an example, the event E_1 of Shankar flying from SF to Washington, D.C. might be described in this calculus as:

$$E_1 \in \textit{Flyings} \wedge \textit{Flyer}(E_1, \textit{Shankar}) \wedge \textit{Origin}(E_1, \textit{SF}) \wedge \textit{Destination}(E_1, \textit{DC})$$
- The predicate $\textit{Happens}(E_1, i)$ could then be used to say this happened during time interval i .
- A time interval has a start and end time (t_1, t_2) .
- Other predicates the event calculus has are:
 - $T(f, t)$ Fluent f is true at time t or interval t
 - $\textit{Happens}(e, i)$ Event e happens over the interval i
 - $\textit{Initiates}(e, f, t)$ Event e causes fluent f to start to hold at time t .
 - $\textit{Terminates}(e, f, t)$ Event e causes fluent f to stop holding at time t .
 - $\textit{Clipped}(f, i)$ Fluent f ceases to be true at some point during the interval i .
 - $\textit{Restored}(f, i)$ Fluent f becomes true sometime during the interval i .
- The book then gives example axioms this calculus is supposed to satisfy. For example, we have the axiom:

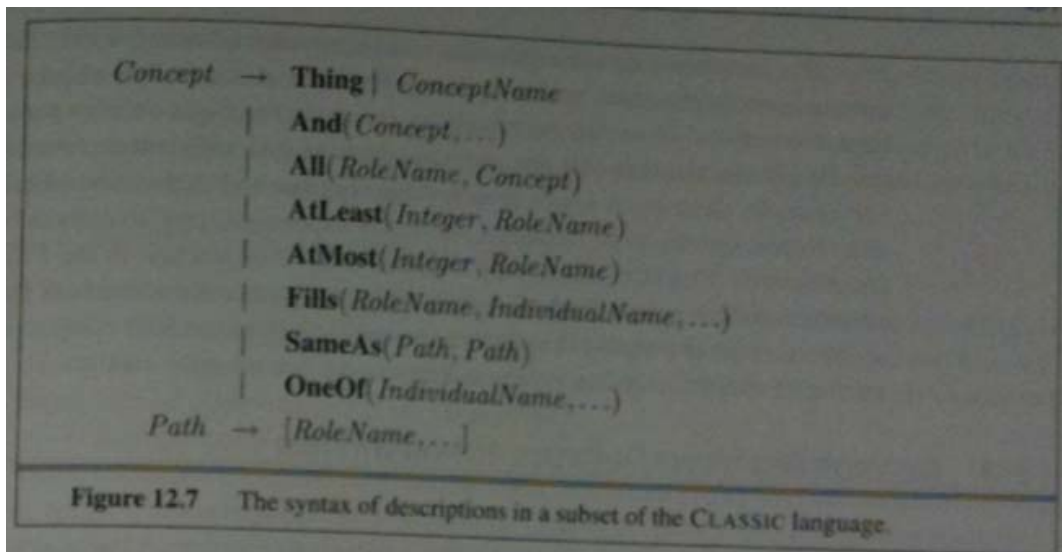
$$T(f, (t_1, t_2)) \Leftrightarrow [\forall t(t_1 < t < t_2) \Rightarrow T(f, t)]$$

Reasoning Systems for Categories



- We would now like describe techniques to integrate reasoning with knowledge systems based on categories
- Two important approaches to this are **semantic networks** and **description logics**.
- We describe the first of these two for the rest of this slide.
- A typical Semantic Network is illustrated above. Objects and Categories are represented with ovals or boxes and connections between them are represented with labeled links. If the link is between objects within categories then it has a double box
- Notice the labelled arcs in the above to nowhere... These are used to represent **default values**.
- Semantic Networks are often used to perform inheritance reasoning based on graph algorithms to determine what properties an object has. Notice they don't easily support n-ary relations so are somewhat limited.

Description Logics



- Description Logics are first-order logics geared towards making it easier to describe definitions and properties of categories.
- They evolved from an attempt to formalize semantics networks more precisely.
- They are geared toward three inferences tasks: **subsumption** - checking if one category is a subset of another, **classification** - checking whether an object belongs to a category, and **consistency** whether the membership criteria of a category is satisfiable or empty. Not all systems can handle support efficient consistency checks.
- The image above is an example description logic called CLASSIC.
- Using CLASSIC we could then represent facts about the world such as: Bachelor = AND(Unmarried, Adult, Male).
- A more complicated example might be:
And(*Man*, *AtLeast*(3, *Son*), *AtMost*(2, *Daughter*),
All(*Son*, *And*(*Unemployed*, *Married*, *All*(*Spouse*, *Doctor*))),
All(*Daughter*, *And*(*Professor*, *Fills*(*Department*, *Physics*, *Math*))))
- Subsumption testing in such a description logic is often p-time and typically involved using unification. i.e., the same techniques as first-order logic but we have weakened the system enough to get better algorithms for subsumption. (The drawback of doing this is it often means hard problems have very long descriptions.)
- Notice DL, doesn't have a notion of negation, and only limited disjunction in Fills and OneOf, this is why it is weaker than usual first-order.

Reasoning With Default Information

- Recall in our Semantic Network example we have a notion of default knowledge. Say that a person has 2 legs.
- This can be overridden for particular instances such as Long John Silver who has one leg.
- Semantic Networks implement the overriding of defaults in a simple and natural way. They don't however give a mechanism for understanding semantics of defaults.
- We would like systems which allow us to model "jumping to conclusions" and such as initially expecting Long John Silver to have two legs and then taking this back.
- Such reasoning violates the monotonicity we have seen in logics so far. So this area is called **nonmonotonic logic**.

Circumscription

- Circumscription is a more powerful version of the closed-world assumption.
- We specify particular predicates that are assumed to be "as false as possible". That is, false for every object except those for which they are known to be true.
- For example, if we want to say by default birds can fly, we introduce a **circumscribed predicate** $Abnormal_1(x)$ and write:
 $Bird(x) \wedge \neg Abnormal_1(x) \Rightarrow Flies(x)$
- A **circumscriptive reasoner** is entitled to assume $\neg Abnormal_1(x)$ unless $Abnormal_1(x)$ is known to be true.
- So we are allowed to conclude $Flies(Tweety)$ from $Bird(Tweety)$ if no other facts are known. However, if somewhere we are given the premise $Abnormal_1(Tweety)$ then we can't derive this conclusion.
- In circumscription, a sentence is entailed if it is true in all preferred models of the KB. Here a model is preferred relative to another if its abnormal objects are subset of the others.

Default Logic

- Default Logic is another example of a non-monotonic reasoning system.
- In default logic we have a sequence of default rules where a rule looks like:

$$\frac{P : J_1, \dots, J_n}{C}$$
 here P is a prerequisite, C is a conclusion and J_i are justifications.
- For example,

$$\frac{Bird(x) : Flies(x)}{Flies(x)}$$
- We say C follows from P unless one of the justifications can be proven false.
- We define the notion of an **extension** of a default theory to be a maximal set of consequences of a theory.
- To determine a maximal set of consequences, we first guess a set of consequences containing the facts, and then see if it can derive back itself using only the facts, and the default rules.
- One common example used in this area is the Nixon diamond which has two extensions:

$$\frac{Republican(Nixon) \wedge Quaker(Nixon). \text{ (fact)}}{Republican(x) : \neg Pacifist(x)} \quad \frac{Quaker(x) : Pacifist(x)}{Pacifist(x)}$$
- One extension is {Republican(Nixon), Quaker(Nixon), Pacifist(Nixon)} and {Republican(Nixon), Quaker(Nixon), not Pacifist(Nixon)}
- If we guess the set {Republican(Nixon), Quaker(Nixon), Pacifist(Nixon)}, then we can derive the first two statements using the fact, and third using the rule on the right.