Logical Generative Models for Probabilistic Reasoning about Existence, Roles and Identity*

David Poole
Department of Computer Science
University of British Columbia
http://www.cs.ubc.ca/spider/poole/

Abstract

In probabilistic reasoning, the problems of existence and identity are important to many different queries; for example, the probability that something that fits some description exists, the probability that some description refers to an object you know about or to a new object, or the probability that an object fulfils some role. Many interesting queries reduce to reasoning about the role of objects. Being able to talk about the existence of parts and sub-parts and the relationships between these parts, allows for probability distributions over complex descriptions. Rather than trying to define a new language, this paper shows how the integration of multiple objects, ontologies and roles can be achieved cleanly. This solves two main problems: reasoning about existence and identity while preserving the clarity principle that specifies that probabilities must be over well defined propositions, and the correspondence problem that means that we don't need to search over all possible correspondences between objects said to exist and things in the world.

Introduction

There has been much recent interest in representations that reason about uncertainty of the existence, identity, types and roles of objects (Koller et al.; 1997; Getoor et al.; 2002; Pasula et al.; 2003; Milch et al.; 2005; Poole and Smyth; 2005; Laskey and da Costa; 2005). Rather than trying to invent a new language, this paper proposes a simple semantic framework that solves many of these issues and can be the target of many languages.

In particular, we aim to integrate

- Existence uncertainty: determining the probability some object that fits some description actually exists (Getoor et al.; 2002; Milch et al.; 2005; Laskey and da Costa; 2005). There has been a long philosophical debate about existence (Miller; 2002), e.g., arguments about whether existence can be a predicate. We do not assume that existence is a predicate.
- Identity uncertainty: determining if two descriptions refer to the same individual or not (Getoor et al.; 2002; Pasula et al.; 2003).
- Type uncertainty: determining the type of an object (or what class it is in) given its properties and properties of other individuals (Koller et al.; 1997).
- Role uncertainty: determining the role of some object, what it can be or will be used for, based on its properties and the properties and roles of other objects[1].
- Probabilities over complex descriptions. For example, a probability distribution over house descriptions where houses can be described at various levels of abstraction (using more general or less general terms) and detail (in terms of what parts and sub-parts are being represented). Poole and Smyth (2005) consider a qualitative probabilistic formulation of this problem (but apply it only to type uncertainty with qualitative probabilities).

In general, this paper should be seen as a contribution to the problem of allowing complex models that simultaneously consider ontologies (that provide, amongst other things, rich type structure), the existence of objects and parts, the roles of objects and identity uncertainty. Some examples of domains where these issues arise are:

- You may want to know the probability that a play would be suitable for your theatre company. The play provides parts, for each of these parts you would like to be able to assign one of the actors. This is a common theme: a model specifies roles for objects in the world to fill.
- A decision may depend on the relationship of parts and other objects. For example, a treatment decision may depend on the location of tumor cells and relationships between them in a cancer patient. A medical system needs to reason about multiple objects that may exist in one patient but not in another patient.
- A real estate agent may want to know the probability that a client would like to be notified of a particular house that came onto the market. The model may specify the probability that the person would like a house given the number of rooms and their features (where rooms that don't exist don't have features), and also depends on the probability that each room in the house can fulfil a role that the person requires.

*This work was supported by an NSERC research grant to the author. Thanks to Peter Carbonetto, Michael Chiang, Mark Crowley, Jacek Kisyński, Kevin Murphy and Rita Sharma for valuable comments.

[1] There are some cases where it is clear whether we are talking about type uncertainty or role uncertainty, but there are many cases where it it is not so clear. The semantics that we give will be able to handle the extremes and those cases where it isn't so clear.

- The tax department may want to know the probability that someone is cheating on their taxes. This requires knowing the claims and their relationships and reasoning about the existence of income that hasn't been declared.

There are a number of desiderata for defining a representation, including:

- Clarity principle: probabilities must be over well-defined propositions. A clairvoyant should be able to determine the truth of any proposition.
- Correspondence problem: if there are multiple objects in the world and multiple names used to describe objects, you don't want to have to reason about all of the combinations of assignments of names to objects. Sometimes the correspondence may be important to answer a query; ideally this would be the only time that the correspondence needs to be explicated.

For example, if there are undifferentiated objects in the world (e.g., the objects in the world are ants that you can't tell apart and can't track), it isn't obvious that a proposition that relies on the correspondence between symbols and objects in the world satisfies the clarity principle. As another example, you may be tempted to have a Boolean random variable $Exists(c_1)$ that is true if c_1 exists. For the case where it is false, you have to be clear what doesn't exist. It doesn't make sense for the object that c_1 refers to to not exist, as then there is no object it refers to. There are some cases where it makes sense, namely when there is only one possible object that could fill the role, such as having a particular arc that doesn't exist in a model (Getoor et al.; 2002), but treating existence as a predicate is not a general solution.

To understand the correspondence problem, consider an example where there are 5 houses, one is observed to be red, and you want to know the probability there is another red house. One possibility is to consider all correspondences of houses to colours, condition on one house being red and measure the probability that one of the other houses is red. Another possibility is that there are 3 hypotheses: there are no red houses, there is one red house, and there is more than one red house. The observation specifies that the first hypothesis doesn't occur, and the problem is to compute the probability of the third hypothesis. While these should give the same answer, the second method lets us reason without thinking about the correspondences.

One solution to both problems has been proposed in the use of Dirichlet processes (Carbonetto et al.; 2005). The general idea is that you need to determine whether each new description refers to one of the objects referred to by a previous description or to a different object. The notion of exchangeability means that it doesn't make sense to worry about the correspondence between objects and names, just whether descriptions denote the same object or different objects. As the only access of the objects is through the descriptions, an actual name-object correspondence wouldn't give us extra capabilities.

This paper extends the idea of Dirichlet processes to allow for logical descriptions, hierarchical types, type uncertainty, objects of different types, objects described at multiple levels of abstraction, and richer observation and query languages.

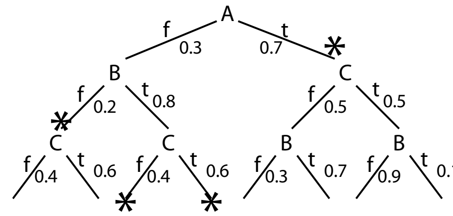

Figure 1: A semantic tree on three Boolean variables. The nodes that determine $a \vee (b \wedge c)$ are starred.

Semantic Tree

We will define the semantics in terms of a semantic tree (this corresponds to the probabilistic part of what decision analysts call a decision tree, or the probabilistic part of a game tree). This is intended to define the semantics, not for computation. This should be seen as a result of a generative model: a path down the semantic tree gives values to a sequence of random variables and specifies the trace of a generative process with its associated probabilities. It does not specify what is observed; this is the role of conditioning that is defined in the standard way.

The simplest case is when there are only discrete variables. In this case, a semantic tree is a tree where each node is labelled with a random variable or some discrete function of a set of random variables. There is an arc for each value of the random variable or each value of the function. For each node labelled with V and each arc labelled with v, there is an associated probability of $V = v$ given the ancestor assignments in the tree. The sum of the probabilities for the children of a node sum to 1. (For completeness you can also assume a set of dummy leaf nodes; these play no role in the description below.)

A path is a set of variable-value pairs in connecting nodes from the root. The probability of a finite path is the product of the probabilities on the arcs along the path. The probability mass of a set of paths, where no path is a sub-path of another path in the set, is the sum of the probabilities of the paths.

We are generally interested in conditional probabilities (the probability of some hypothesis given evidence). $P(h|e) = P(h \wedge e)/P(e)$. Thus all we need to define is the probability of propositions and it can be extended to conditional probabilities.

A proposition defined on variables is **determined** at some node if the truth of the proposition can be evaluated just based on the assignment associated with the path to that node. A node **determines** a proposition if the proposition is determined at that node, and isn't determined at any ancestor node. The probability of a proposition is the probability mass of the paths to the nodes that determine the proposition to be true.

Example 1 Figure 1 shows a semantic tree for three Boolean variables, A, B and C (where a means $A = true$, etc.). The nodes that determine the proposition $a \vee (b \wedge c)$ are starred. At the path $[A = f, B = t, C = t]$ the proposition is true and at the path $[A = f, B = f]$ the proposition is false.

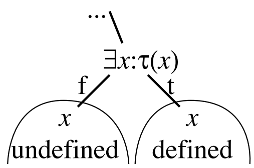

Figure 2: Splitting on existence

When the semantic tree is infinite (e.g., there infinitely many random variables, or we are modelling finer discretizations of a continuous variable), propositions are only defined on finite sub-trees. The probability of a proposition is well defined if, for every $\epsilon > 0$, there is a finite sub-tree in which the probability mass of the paths to the nodes that determine the proposition is greater than $1 - \epsilon$. The probability of the proposition is the limit, as ϵ goes to zero, of the probability mass of the paths to the nodes that determine the proposition to be true in these finite sub-trees. Intuitively ϵ is the error that you obtain by restricting to a finite tree. To make sure that this is well defined, we need a notion of **fairness**: any variable that can be chosen down a path to split on is eventually chosen. For example, if we have two infinite sequences of variables X_i and Y_i, we can't build a semantic tree by only choosing the X_i's and never splitting on the Y_i's.

This notion of a semantic tree can be used to define the limiting process that gives the standard definition of a measure over continuous variables: moving down the tree can provide a finer partition of the continuous variables.

Note that any semantic tree can be converted into a Boolean semantic tree (with binary true/false splits) by recursively splitting the domain of discrete variables.

Existence and Ontologies

To add existence to the semantic tree, we split on whether an object exists. In particular, we extend the language used in a semantic tree to the first-order predicate calculus. Splitting on existence will introduce scoping of variables.

It doesn't really make sense to specify the probability that something exists without giving it a type or a description. In any realistic domain, "something else" always exists. If you didn't want to explicitly type individuals, you would need to implicitly type them (to those individuals you are modelling).

We can split on the existence of an object of a certain type with certain properties. The simplest case is to split on

$$\exists x\ \tau(x)$$

where τ is a type (a predicate on individuals). The existence is either true or false. If it is false, no such individual exists. If it is true, such an object exists, and we can use x to refer to one such object (see below). The semantics is defined below so that properties of x are only defined in the sub-tree under the branch where existence is true. See Figure 2. Note that in the sub-tree under the branch where $\exists x\ \tau(x)$ is true, we know there is one or more objects of type τ.

A more general case is to split on

$$\exists x\ \tau(x) \wedge q(x, \overline{y})$$

where $\overline{y}$ is a set of the names of other individuals. The elements of $\overline{y}$ must have been defined by ancestors in the semantic tree (i.e., $q(x, \overline{y})$ must be in the scope of a quantification on $\overline{y}$). In the case that there is more than one such x, this does not specify which of the objects x refers to; it is just true if there exists some x. This issue is discussed in more detail below.

We assume that the types are put in some hierarchy using an ontology. There is much current work on ontologies (McGuinness and van Harmelen; 2004; Patel-Schneider et al.; 2004), part of which involves allowing the hierarchical description of types of objects (the type of an object is a class, and classes can be hierarchically organized). In a manner similar to P-Classic (Koller et al.; 1997), we recursively split into subtypes, where the probability distribution of the subtypes can be a function of any of the ancestors in the semantic tree.

We assume that the world contains objects of the lowest level of the abstraction hierarchy (generic buildings don't exist; only houses, apartment buildings, office buildings, etc. exist). These lowest level types we call *primitive types*. In a semantic tree, in every branch under where $\exists x\ \tau(x) \wedge q(x, \overline{y})$ is true, if τ is not a primitive type, the type of x must be split into its subtypes. Each of these subtypes must be split into their subtypes until only primitive types remain. This is important for fairness defined below.

Multiple Objects

Suppose the domain we want to model consists of a population of individuals. There are two ways to model this, the first is to branch on the number of individuals of some type. The second is to split on each individual that exists. Both methods have their semantic challenges.

If one first splits on the number of individuals, the node is the number of individuals (of a particular type) and the arcs are labelled with integers. This is challenging for a number of reasons, the main ones being:

- The individuals that exist under the different branches are unrelated semantically. An individual that exists when there are 5 individuals may not exist when there are 6. This means that you cannot ask about a probability that refers to a particular individual. It also is not clear how to refer to a particular role of an individual or how to select particular individuals.
- The number of individuals may depend on the type (or the role) of other individuals that exist. For example, the number of rooms in a house is a function of their properties; it may be unlikely to have four bedrooms if there is only one bathroom. As another example, dining room chairs often come in sets of four or six identical (or closely related) chairs, but if the chairs are all different, then there is no reason to think that a groups of five chairs should be treated differently than groups of four or six chairs.

Splitting on the number of individuals is being pursued in BLOG (Milch et al.; 2005), where these issues may or may not become a problem.

This paper explores the second method. Given some objects, we split on whether there is one more. This lets us refer to the objects individually (by giving them names when

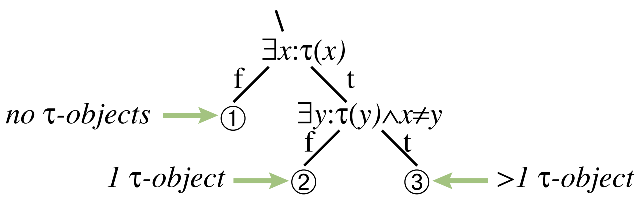

Figure 3: Existence of zero, one or more objects

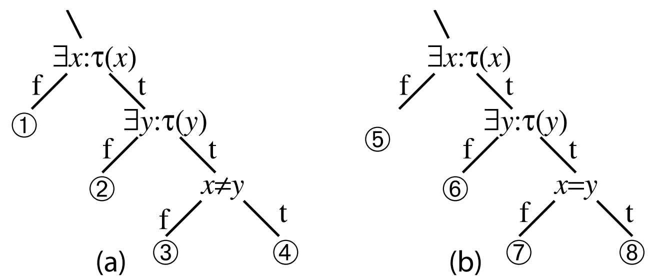

Figure 4: An example showing asymmetry between true and false

they are introduced). It also lets us condition the existence of the next object on the properties of the preceding objects.

For example, suppose we have $\exists y\ \tau(y) \wedge y \neq x$ under $\exists x\ \tau(x)$ in a semantic tree (see Figure 3). In this case, under the branch where $\exists y\ \tau(y) \wedge y \neq x$ is false, we know there is exactly one object of type τ. We can refer to it as x; y is not defined here. Under the branch where $\exists y\ \tau(y) \wedge y \neq x$ is true there are at least two objects of type τ.

Semantics

A possible world is defined in a standard manner as a set of individuals (the domain) and an assignment of a relation to each predicate symbol. The first order semantic tree defines a measure over those sets of possible worlds that can be described by first-order sentences. We allow the worlds to be heterogeneous in that they do not need to share the same domain. We assume that all constants and functions are defined by existential statements (by describing the individuals said to exist). This correspond to the use of Skolem constants. (For simplicity this papers ignores Skolem functions with parameters or, equivalently, existential quantification in the scope of universal quantification.)

Formally, a **first-order semantic tree** is a (possibly infinite) binary tree with a first-order formula at every interior node, where the children are labelled true and false (we put the true branch on the right), and a probability distribution over the children of each node. Probabilities can be defined arbitrarily as long as 0 is associated with logical contradiction[2]. A formula can contain a free variable as long as it is a descendant of a "true" branch of a node that quantifies that variable.

Reading down the branches to a node gives a logical formula. Each path from the root corresponds to a formula. We recursively define the **path formula** to node n, written as $pf(n)$ as follows: (Note that the path formula to node n does not take into account the formula at n). The path formula of the root node is "*true*". If the path formula of node n is formula f (i.e., $f = pf(n)$) and node n is labelled with formula f', the "true" child of node n has path formula $f \wedge f'$, where f' is in the scope of the quantification of f. The path formula to the "false" child of node n is:

$$f \wedge \neg(f \wedge f')$$

This says that the path formula into n holds and the true branch out of n doesn't hold.

A **logical generative model** is a model in some language that constructs a first-order semantic tree.

Example 2 In Figure 3, the path formula to node ③ is

$$\exists x\ \tau(x) \wedge \exists y\ \tau(y) \wedge x \neq y$$

which is true when there are two or more objects of type τ. Any descendants of this node are in the scope of x and y.

Node ② has path formula:

$$\exists x\ \tau(x) \wedge \neg(\exists x\ \tau(x) \wedge \exists y\ \tau(y) \wedge x \neq y)$$

which is true when there is exactly one object of type τ. Note that any descendants of this node are in the scope of x, but not in the scope of y.

The following example shows why the false branch doesn't correspond to $f \wedge \neg f'$.

Example 3 Consider the two semantic trees of Figure 4. While it may seem that they mean the same (swapping the bottom-right branches), they have different meanings:

In Figure 4(a), node ④ has path formula

$$\exists x\ \tau(x) \wedge \exists y\ \tau(y) \wedge x \neq y$$

which is true when there are two or more objects. Node ③ has path formula

$$\exists x\ \tau(x) \wedge \exists y\ \tau(y) \wedge \neg(\exists x\ \tau(x) \wedge \exists y\ \tau(y) \wedge x \neq y)$$

which is true when there is exactly one object of type τ. Node ② has path formula *false*, since if there exists an x of type τ, there must exist a y of type τ, as $\exists x\ \tau(x) \rightarrow \exists y\ \tau(y)$.

In Figure 4(b), node ⑧ has path formula

$$\exists x\ \tau(x) \wedge \exists y\ \tau(y) \wedge x = y$$

which is logically equivalent to $\exists x\ \tau(x)$. Thus nodes ⑥ and ⑦ both have path formula *false*.

We can have quite complicated conditions of existence to represent what different objects exist:

Example 4 Suppose we want to represent an urban block of land that usually contains a park or two houses or a single building that is not a house. Other combinations are possible, but less likely. In this example, we could split on the existence of any of these, but consider the case where we split

[2] This means that deciding the consistency of a first-order semantic tree is undecidable, in general. Undecidability is a consequence of having an expressive language. As Gödel showed (Nagel and Newman; 2001), any language capable of representing arithmetic is undecidable. If you cared, you could restrict the language to be decidable. However, that would just make the language less expressive, not more efficient. A more intriguing possibility is to remove the restriction on inconsistent sentences having a prior probability of zero; this would enable one to have the probability that some proposition is a tautology (e.g., for betting on $P = NP$). We could then condition on proofs. But that leads us too far afield from this paper.

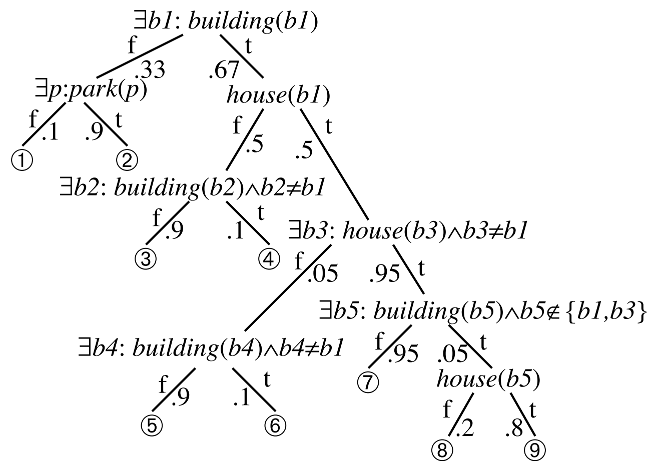

Figure 5: A semantic tree for what is on a block of land

on whether a building exists or not, then, when it exists, we split on whether it is a house. A portion of one such tree[3] is shown in Figure 5. The circled numbers represent positions at which some propositions could be resolved; there are possibly infinite sub-trees at these positions that are not shown.

Nodes ②, ③ and ⑦ correspond to the three normal situations, each with a probability about 0.3. Node ② corresponds to the park with no buildings. Node ③ corresponds to the case where there is exactly one building that isn't a house.

Node ⑦ corresponds to the case where there are two houses and no other buildings. That is, it corresponds to all of the worlds where:

$$\exists b_1\ building(b_1) \wedge house(b_1) \wedge \exists b_3\ house(b_3) \\ \wedge\ b_3 \neq b_1 \wedge \neg(\exists b_5 building(b_5) \wedge b_5 \notin \{b_1, b_3\})$$

The set of these worlds has probability measure $0.67 * 0.5 * 0.95 * 0.95$.

Our semantics of the probability of a proposition is the same as before: considering the nodes that determine the proposition, the probability of the proposition is the sum of the probabilities of the paths to those nodes where the proposition is true.

More formally, a proposition α, that is represented by a closed formula, is **determined** at a node *n* if the path formula to node *n* entails α or entails $\neg\alpha$. That is, if $pf(n) \models \alpha$ or $pf(n) \models \neg\alpha$. A node *n* **determines** α if α is determined at *n*, but isn't determined at any ancestor of *n*.

Given a finite semantic tree *T* and a proposition α, define $U_T(\alpha)$ to be the sum of the probabilities of the paths to the leaves of *T* where α is not determined. Define

$$P_T^+(\alpha) = \sum_{L \text{ is a leaf of } T \text{ and } pf(L) \models \alpha} P(L)$$

where $P(L)$ is the product of the probabilities on the branches to leaf node *L*. Let $P_T^-(\alpha) = 1 - P_T^+(\neg\alpha)$. It is straightforward to prove that $U_T(\alpha) + P_T^+(\alpha) + P_T^-(\alpha) = 1$.

A (possibly-infinite) semantic tree *T* is **fair** with respect to proposition α if for every $\epsilon > 0$ there exists a finite sub-tree T' of *T* such that $U_T(\alpha) < \epsilon$.

We can define the probability in an infinite semantic tree as a limit: $P_T(\alpha) = p$ if for every $\epsilon > 0$, there is finite subtree tree T' of *T* such that $P_{T'}^+(\alpha) - p < \epsilon$. The fairness guarantees that the limit exists.

Note that the individuals that exist may be different depending on the path. That is, if a query requires some individuals to exist (e.g., we have observed the existence of some individual), the correspondence between the individuals that exist in the tree and in the proposition can be different for different paths.

Example 5 Suppose we are given the tree of Example 4 and the query: *What is the probability that there are at least two buildings, one of which is not a house?* This is:
$P(\exists x\ building(x) \wedge \exists y\ building(y) \wedge x \neq y \wedge \neg house(x))$
To determine this probability, we trace down the tree, finding the positions where there are two buildings, one of which is not a house. In Figure 5, this proposition is true at position

- ④, where $b1$ fulfils the role of the non-house; $b2$ is a building that could be a house (i.e., x in the query corresponds to $b1$ and y corresponds to $b2$).
- ⑥, where $b4$ fulfils the role of a non-house, and $b1$ is the other building (in this case a house). Note that we know that $b4$ is not a house as it is is false that there exists a house that isn't $b1$.
- ⑧ where $b5$ is the non-house, and either $b1$ or $b3$, both of which are houses, could fulfil the role of the other building.

This proposition is not determined at position ⑨. The probability of the branch to ⑨ is 0.013, so if you wanted probabilities more accurate than that, you would need to expand the tree more.

The above tools are enough to have a distribution over the number of objects, as long we can refer to the set of previous existing objects.

Example 6 Suppose we want to represent that dining chairs come in sets of 4 or 6. The distribution over the number of chairs should reflect this. Note that this only affects identical chairs. Let's first split on

$$\exists x_1\ chair(x_1)$$

This is true if there is a chair. When it is true, we can split on

$$\exists x_2\ chair(x_2) \wedge x_2 \notin \{x_1\} \wedge identical(x_2, x_1)$$

This is true when there are two identical chairs. The probability that this is true might be high. When it is true, we can model the third chair by splitting on

$$\exists x_3\ chair(x_3) \wedge x_3 \notin \{x_1, x_2\} \wedge identical(x_3, x_1)$$

Again the probability that this exists is high. We can do the same for the rest of the chairs in the set. The probability of the fourth chair would be high, but the probability of the fifth chair would be low. The probability of the sixth chair would be high. The probability of the seventh chair would be low. We could place an exponential distribution over the

[3]This is not intended to be the most straightforward tree, but is designed to show a possible tree that someone could represent for this example. Our systems have to cope with all representations given by users. We have to be able to interpret all of the legal trees.

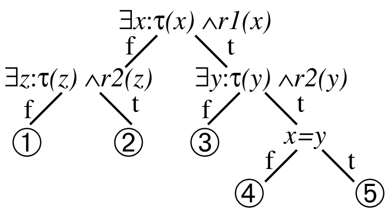

Figure 6: Reasoning about roles and identity

existence of the remaining chairs by making the existence of each new chair some constant. We can also model the existence of non-matching chairs on the sub-trees when these propositions are false.

Note that we did not need to do the combinatorial matching of different chairs to the labels x_1, x_2, etc. We are only doing a decision problem down each branch.

Roles and Identity

In the previous examples, the condition specifying the existence of a new object stated that it must be different to previous objects. This is not adequate to handle identity uncertainty (determining if two descriptions refer to the same individual or to different individuals), or the problem of role identification described below.

Often when we want to model the existence of an object, we want to hypothesize whether there is an object that fulfils a role. That is, we may hypothesize that an object exists that fulfils a role, but not know whether the object that exists is an object we already know about (i.e., is defined as an ancestor in the semantic tree) or is a new object. Moreover there may be multiple objects that could fulfil a role and one object could potentially fulfil many roles.

We do not need to actually make a definition of a role. If we can state properties of the individual that fills a role, the role assignment can be done probabilistically. Individuals that fit the role description will be more likely than those who do not.

Example 7 Figure 6 shows a typical scenario for reasoning about roles and identity. First we split on the existence of an object to fulfil role $r1$:

$$\exists x\ \tau(x) \wedge r1(x)$$

If this is true then there is something that fills role $r1$. We can then split on the existence of an object that fills role $r2$. If there are objects that fill both roles, we can split on whether they are the same object.

In this semantic tree (imagine that it is part of a bigger tree), there are 5 qualitatively different regions:

- at ① there are no objects that fill either role.
- at ② there is an object that fills role $r2$ but no object that fills role $r1$.
- at ③ there is an object that fills role $r1$ and no object that fills $r2$
- at ④ only different objects fulfil the two roles
- at ⑤ some object or objects fills both roles

Note that equality is only defined when both objects exist. It is undefined in the other parts of the sub-tree. We need to be careful about what we mean by equality if we do not condition on the existence of the objects. Typically, $P(x = y) + P(x \neq y) < 1$ as these are not the only hypotheses. The sum only equals 1 when there is probability 1 that there exist objects that fill the roles.

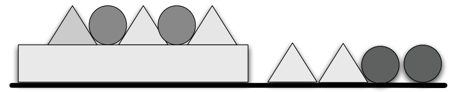

Figure 7: World with multiple objects

Exchangeability

The above framework works fine for computing conditional probabilities, using $P(h|e) = P(e \wedge h)/P(e)$ as long as h does not refer to a logical variable in e. In that case, we only need to ask decision questions, and there is no need for enumerating correspondences.

There is a class of queries that requires one more construct; this is when we want to ask a question about an individual that we have ascertained exists. The evidence e may have included $\exists x\ c(x)$, and thus we have observed that some x exists. The query h may want to know something about the x that exists; the problem is to determine which x. The notion of **exchangeability** (Gelman et al.; 2004) specifies that a priori each individual is equally likely to be chosen. This is similar to the idea of choosing a random individual (Bacchus; 1990).

Example 8 Suppose we have the the domain of Figure 7 (assume either there is only one possible world or we have previously conditioned so that this is the only possible configuration of shapes) and observed that a triangle is touching a circle, and we want to know the probability it (the triangle) is on a rectangle. The observation is

$$\exists x\ \mathit{triangle}(x) \wedge \exists y\ \mathit{circle}(y) \wedge \mathit{touching}(x, y).$$

The query q is $\exists z\ \mathit{rectangle}(z) \wedge \mathit{on}(x, z)$, where the x that is free in the query is intended to refer to the triangle in the observation. We can't just add the query into the scope of x forming $e \wedge q$ as

$$\exists x\ \mathit{triangle}(x) \wedge \exists y\ \mathit{circle}(y) \wedge \mathit{touching}(x, y) \wedge \exists z\ \mathit{rectangle}(z) \wedge \mathit{on}(x, z).$$

because this is true and so has probability 1 (note that using the same technique $e \wedge \neg q$ is also true, where different individuals are chosen for x). What we want to do here is to ask whether a random x which fits the observation is on the rectangle. This involves counting which proportion of the individuals that we have ascertained exist satisfy the query.

To handle these cases where we want to refer to a particular individual that has been chosen or observed, we can extend the notion of a semantic tree. A **generalized first-order semantic tree** is a first-order semantic tree that can contain a $\mathit{commit}(\bar{x})$ node where $\bar{x}$ is a set of variables in whose scope this node is, that are not in an ancestor commit. This node has one child.

To define the semantics, we need to specify which possible worlds get mapped to each path in the semantic tree. At the $\mathit{commit}(\bar{x})$, for each possible world in which the path formula to the commit is true, we assign a particular tuple of individuals to the variables $\bar{x}$. (I.e., it is as though we had $x_i = c_i$, where c_i denotes a particular individual, for each x_i in $\bar{x}$). We assume that each individual in the possible world that satisfies the path formula to the commit node has an equal chance of being chosen.

The commit can be used to represent the protocol of how the individual was selected. Continuing Example 8, there a number of ways that the individuals can be chosen. Suppose we first select a circle at random, then observe that it is touching a triangle. This can be represented as committing to y before we commit to x. In this case, the probability that x is on some z is 2/3. If we first selected a triangle at random, then observed that it is touching a circle, this can be represented as committing to x before committing to y in the tree. In this case, the probability that x is on some z is 3/4. If we commit to x and y simultaneously, there are 4 x-y pairs where x is on a rectangle and one that isn't, so the probability that x is on some z is 4/5.

Conclusions

This paper has shown how to integrate existence, identity, roles and ontologies into a clean semantic framework. All probabilities are over well-define propositions, and we have avoided the issues that lead to the slogan "Existence is not a predicate" (Miller; 2002), and avoided much of the combinatorial assignment of constants to individuals. For some queries, however, we need to determine whether one description refers to the same individual as another description, and we have identified cases where reasoning about existence alone is not adequate and we need to reason about randomly chosen individuals.

This paper originated in trying to design languages to reason about complex domains with multiple objects described using rich ontologies (e.g., Poole et al.; 2007; Sharma et al.; 2007). It soon became clear that we needed reason about the probability of existence, and to do this we need to be very careful about exactly what doesn't exist when the existence is false. We also wanted to reduce the combinatorial explosion of assignments of descriptions to individuals that has the potential to make reasoning infeasible. These issues transcend any particular language. To use the results of this paper, you need to design a language that constructs a generalized first-order semantic tree. This is non-trivial as when there is no a priori bound on the number of individuals, the semantic tree will be infinite, and we need a finite representation of the infinite tree. There are many possible languages. In general we want a compact representation that can generate the appropriate subset of an infinite semantic tree to answer a particular query. One possibility is to use the idea of Bayesian networks, where the probability depends on a finite set of ancestors in the tree (corresponding to the parents in a Bayesian network). To make this work, we need to be able to access the names of the individuals existing in a path, to be able to say that one more individual exists.

References

Bacchus, F. (1990). *Representing and Reasoning with Uncertain Knowledge*, MIT Press, Cambridge, Massachusetts.

Carbonetto, P., Kisynski, J., de Freitas, N. and Poole., D. (2005). Nonparametric Bayesian logic, *UAI-2005*, Edinburgh.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (2004). *Bayesian Data Analysis*, 2nd edn, Chapman and Hall/CRC.

Getoor, L., Friedman, N., Koller, D. and Taskar, B. (2002). Learning probabilistic models of link structure, *Journal of Machine Learning Research (JMLR)* **3**: 679–707.

Koller, D., Levy, A. and Pfeffer, A. (1997). P-classic: A tractable probabilistic description logic, *AAAI-97*, Providence, RI, pp. 390–397.

Laskey, K. B. and da Costa, P. G. C. (2005). Of klingons and starships: Bayesian logic for the 23rd century, *Uncertainty in Artificial Intelligence: Proceedings of the Twenty-First Conference*.

McGuinness, D. L. and van Harmelen, F. (2004). Owl web ontology language overview, *W3C Recommendation 10 February 2004*, W3C. http://www.w3.org/TR/owl-features/

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L. and Kolobov, A. (2005). BLOG: Probabilistic models with unknown objects, *IJCAI-05*, Edinburgh.

Miller, B. (2002). Existence, *in* E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. http://plato.stanford.edu/archives/sum2002/entries/existence/

Nagel, E. and Newman, J. R. (2001). *Gödel's Proof*, revised edition edn, New York University Press.

Pasula, H., Marthi, B., Milch, B., Russell, S. and Shpitser, I. (2003). Identity uncertainty and citation matching, *NIPS*, Vol. 15.

Patel-Schneider, P. F., Hayes, P. and Horrocks, I. (2004). Owl web ontology language: Semantics and abstract syntax, *W3C Recommendation 10 February 2004*, W3C. http://www.w3.org/TR/owl-semantics/

Poole, D. and Smyth, C. (2005). Type uncertainty in ontologically-grounded qualitative probabilistic matching, *Proc. European Conference on Symbolic and Qualitative Reasoning About Uncertainty (ECSQARU)*, LNAI 3571, Springer-Verlag, Barcelona, Spain, pp. 763–774.

Poole, D., Smyth, C. and Sharma, R. (2007). Ontologically-grounded probabilistic modelling, *submitted*.

Sharma, R., Poole, D. and Smyth, C. (2007). A system for ontologically-grounded probabilistic matching, *submitted*.