ProbSem: A Probabilistic Semistructured Database Model

Edward Hung

Department of Computer Science,
University of Maryland, College Park, MD 20742.
ehung@cs.umd.edu

Abstract

Recent interest in semistructured data has led to a proliferation of XML-based standards which encompass applications ranging from multimedia applications and sensor data processing applications to financial applications and myriads of other more traditional applications. When semistructured paradigms are used to store sensor data and multimedia (e.g. image) data, we need to be able to handle uncertainty as sensor readings and image processing methods often yield uncertain results. In this paper, we propose the concept of *probabilistic semistructured (PS) databases*. We propose a *global* notion of consistency as well as a *local* one and show that the two coincide.

1 Introduction

The semistructured model of data has come a long way since the introduction of the OEM model [8]. The popular XML markup language (and its numerous variants which address a huge diversity of application sectors) is based on the semistructured paradigm. XML variants exist for a wide variety of multimedia data representation and reasoning tasks as well as for a wide variety of sensor processing tasks. Both these domains are fraught with uncertainty. For example, statistical models for object recognition in images have a long history. In the same vein, statistical models for assessing the reliability of sensor readings are numerous. Yet, a formal model of semistructured data that accounts for uncertainty is yet to be developed (though a model that allows random variables to be stored using semistructured databases has been proposed by Dekhtyar et al[3]). Our proposal does the opposite - it extends the semistructured data model so that paths in such a model can include probabilistic information.

The primary contributions of this paper are as follows. We first start with a motivating example in Section 2. In Section 3, we propose the ProbSem model of PS-databases based in part on graphs (as in standard semistructured data models). Then, in Section 4, we define two semantics for PS-databases. The first is a *global* semantics (in a sense to be made precise) while the second is a *local* semantics. Both semantics turn out to be equivalent which means that instead of reasoning at a "global" level, our algebra can reason on a local level (i.e. on a per node basis for nodes in the graph). We prove that according to both these semantics, every PS-database is guaranteed to be consistent.

2 Motivating Example

Consider a surveillance application where a battlefield is being monitored. Image processing methods are used to classify objects appearing in images. Some objects are classified as vehicle convoys or refugee groups. Vehicle convoys may be further classified into individual vehicles. However, there may be uncertainty over the number of vehicles in the convoy: we may have an estimate that the convoy contains 10 to

15 vehicles. These vehicle objects may be further classified into categories such as tanks, cars, armored personnel carriers. Again there maybe uncertainty as to the categorization of a vehicle: we may think it is likely to be a tank, but there is some chance it is a (small) armored personnel carrier.

In such environments, uncertainty abounds. We have already seen the existence of uncertainty above in terms of the number of vehicles in a vehicle convoy and the classification of the vehicles. Further uncertainty may arise because image processing methods may not explicitly extract the identity of the objects. For instance, even if we know for sure that a given vehicle is a tank, we might not be able to classify it further into a T-72 tank or a T-80 tank. Depending on the resolution of the imagery, even further classifications may be possible.

Semistructured data is a natural way to store such data because for a surveillance application of this kind, we have some idea of what the structure of data looks like (e.g. the general hierarchical structure alluded to above). Unfortunately, to date, little work has been done on developing data models to store uncertain information in probabilistic environments, which our work is focused at.

3 Probabilistic semistructured (ProbSem) Data Model

In this section, we introduce the ProbSem probabilistic semistructured data model. We start by introducing syntactic definitions of this data model, and then in Section 4, we introduce two formal model theories for PS-databases.

We first review the definition of a semistructured data (SD) data model and then we will introduce the syntax for the ProbSem probabilistic semistructured data model.

3.1 Semistructured Data Model

We start by recalling some simple graph concepts.

Definition 3.1 *Let V be a finite set (of vertices), $E \subseteq V \times V$ be a set (of edges) and $\ell : E \rightarrow \mathcal{L}$ be a mapping from edges to a set $\mathcal{L}$ of strings called* labels *. The triple $G = (V, E, \ell)$ is a* **rooted, edge labeled directed graph**.

As usual, a graph is *rooted* iff there is a distinguished node called the root such that for every node in the graph, there is a path in the graph from the root to that node.

Definition 3.2 *Suppose $G = (V, E, \ell)$ is any rooted, edge-labeled directed graph. For $o \in V$:*

- *The* **children** *of o is the set $\mathsf{C}(o) = \{o' \mid (o, o') \in E\}$.*
- *The descendants of o is the set $\mathsf{des}(o) = \{o' \mid o' \in V \wedge (o' \in \mathsf{C}(o) \vee \exists o'' \in V(o'' \in \mathsf{C}(o) \wedge o' \in \mathsf{des}(o'')))\}$.*
- *The non-descendants of o is the set $\mathsf{ndes}(o) = \{o' | o' \in V \wedge o' \notin \mathsf{des}(o)\}$.*
- *We use $\mathsf{lch}(o, l)$ to denote the set of children of o with label l. More formally,*

$$\mathsf{lch}(o, l) = \{o' \mid \wedge(o, o') \in E \wedge \ell(o, o') = l\}.$$

- *The parents of o, $\mathsf{parents}(o)$, is the set $\{o' \mid (o', o) \in E\}$.*
- *A vertex o is called a* **leaf** *iff $\mathsf{C}(o) = \emptyset$.*

Ancestors and descendants of nodes can be defined in the usual way. It is important to note that our graphs can include cycles - this is particularly important in the semistructured data model as two objects may refer to each other.

As we plan to build upon existing models of semistructured databases, we start by recapitulating the definition of a semistructured instance from [1]. We start by assuming the existence of some arbitrary but fixed set $\mathcal{O}$ of strings called object-ids (oids for short), and a set $\mathcal{T}$ of types. Each type $T \in \mathcal{T}$ has an associated finite domain, $\mathsf{dom}(T)$.

Definition 3.3 *A* **semistructured instance** *$\mathcal{S}$ over a set of objects $\mathcal{O}$, a set of labels $\mathcal{L}$, and a set of types $\mathcal{T}$ is a 5-tuple $\mathcal{S} = (V, E, \ell, \tau, \mathsf{val})$ where:*

1. *$G = (V, E, \ell)$ is a rooted, directed graph where $V \subseteq \mathcal{O}$, $E \subseteq V \times V$ and $\ell : E \to \mathcal{L}$;*
2. *τ associates a type in $\mathcal{T}$ with each leaf object o in G.*
3. *val associates a value in $\mathsf{dom}(\tau(o))$ with each leaf object o.*

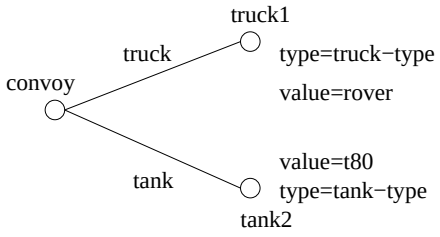

Figure 1: A semistructured instance for the surveillance domain.

We illustrate the above definition through an example from the surveillance domain.

Example 3.1 *Figure 1 shows a graph representing a part of the surveillance domain.*

- *The instance is defined over the set $\mathcal{O} = \{convoy, truck1, truck2, tank1, tank2\}$ of objects. The set of labels is $\mathcal{L} = \{truck, tank\}$. There are two types, truck-type and tank-type. with domains given by: dom(truck-type) $= \{mac, rover\}$ and dom(tank-type) $= \{t70, t72, t80\}$.*
- *The graph shows that the instance consists of 3 objects: $convoy$, $truck1$ and $tank2$.*
- *The instance includes the edges $(convoy, truck1)$ and $(convoy, tank2)$, labelled with $truck$ and $tank$ respectively.*
- *Finally, the types and values of the leaves in the instance are: $\tau(truck1)$ = truck-type, $\tau(tank2)$ = tank-type, $\mathsf{val}(truck1) = rover$ and $\mathsf{val}(tank2) = t80$.*

3.2 The ProbSem Probabilistic Data Model

In this section, we develop the basic syntax of the ProbSem probabilistic data model. However, before defining the important concept of a probabilistic instance, we need some intermediate concepts.

Definition 3.4 *Suppose S is any set. An* **interval function** *ι w.r.t. S associates, with each $s \in S$, a closed subinterval $[lb(s), ub(s)] \subseteq [0, 1]$. ι is called an* **interval probability function** *if it satisfies the following additional conditions:*

1. $\sum_{s \in S} lb(s) \leq 1$.

2. $\sum_{s \in S} ub(s) \geq 1$.

An interval function merely assigns subintervals of $[0, 1]$ to members of the set S. However, an interval *probability* function has the additional requirement that these assignments make sense from a probabilistic point of view. Intuitively, think of S above as being the set of all possible outcomes of some event and think of $\iota(s) = [lb(s), ub(s)]$ as saying that according to the interval probability function ι, the probability that s is the actual outcome lies in the range $[lb(s), ub(s)]$.

Definition 3.5 (probability distribution) *A probability distribution w.r.t. S over an interval probability function ι is a mapping $\mathcal{P} : S \rightarrow [0, 1]$ where*

1. $\forall s \in S, lb(s) \leq \mathcal{P}(s) \leq ub(s)$.

2. $\Sigma_{s \in S} \mathcal{P}(s) = 1$.

Lemma 1 *For any set S and any interval probability function ι w.r.t. S, there exists a probability distribution P(S) which is compatible with ι.*

Proof: There are potentially many possible distributions that are compatible with ι. One solution is the distribution that is as close to "in the middle" of each interval as possible.

Let $\sum_{s \in S} lb(s) = L$ and $\sum_{s \in S} ub(s) = U$. By the defintion of interval probability function we know that $L \leq 1$ and $U \geq 1$. A probability function that is consistent with the interval constraints is:

$$p(s_i) = lb(s_i) + (ub(s_i) - lb(s_i)) \times \frac{1 - L}{U - L}.$$

It is easy to check that $\sum_i p(s_i) = 1$ and that $lb(s_i) \leq p(s_i) \leq ub(s_i)$. ∎

Example 3.2 *Suppose we have a convoy that is known to contain a tank, but we do not know if it is a T-70, T-72, or T-80 tank. We use 3 tank-type objects $tank1$, $tank2$, $tank3$ with values of $t70, t72, t80$ respectively to denote the three possibilities. Then, our set $S = \{tank1, tank2, tank3\}$. The function ι which assigns $[0.2, 0.4]$ to $tank1$, $[0.5, 0.6]$ to $tank2$ and $[0.1, 0.2]$ to $tank3$ is an interval probability function because $0.2 + 0.5 + 0.1 = 0.8$ which is less than or equal to 1 and $0.4 + 0.6 + 0.2 = 1.2$ which is greater than 1. The construction in lemma will assign 0.3 to $p(tank1)$, 0.55 to $p(tank2)$ and 0.15 to $p(tank3)$.* [1]

Note: Given an interval $I = [x, y]$ we will often use the notation $I.lb$ to denote x and $I.ub$ to denote y.

Interval probability functions may assign extremely loose bounds which can often be tightened. For example, if $S = \{s_1, s_2\}$ and $\iota(s_1) = [0.5, 0.7], \iota(s_2) = [0.2, 1]$ then the upper bound of $\iota(s_2)$ is loose - we can easily tighten it to 0.5. Likewise, the lower bound of $\iota(s_2)$ is also loose - it can be tightened to 0.3. We now define the tightness of an interval probability function, a tightening operator and the corresponding theorem by using Definition 8, 9, 10, 11 and Theorem 2 of [3], which are rewritten below.

Definition 3.6 *Suppose ι is an interval probability function w.r.t. S. ι is* **tight** *iff for any interval probability function ι' w.r.t. S such that every probability distribution $\mathcal{P}$ over ι is also a probability distribution over ι', $\iota(s).lb \geq \iota'(s).lb$ and $\iota(s).ub \leq \iota'(s).ub$ where $s \in S$. If every probability distribution P over ι' is also a probability distribution over ι, then we say that ι is the* **tight equivalent** *of ι'.*

[1] This is really a conditional distribution; it is conditioned on the fact that the convoy contains a tank. In the next section, we will describe more precisely how this is done.

Definition 3.7 *A* **tightening operator** *is a mapping* tight *from interval probability functions to interval probability functions such that* $\mathsf{tight}(\iota)$ *is always tight.*

The following theorem says that we can tighten a given interval probability function by some simple arithmetic calculations.

Theorem 1 *Suppose* ι, ι' *are interval probability functions over* S *and* $tight(\iota) = \iota'$*. Let* $s \in S$*. Then:*

$$\begin{aligned} \iota'(s) \;=\; & [max(\iota(s).lb, 1 - \sum_{s' \in S \wedge s' \neq s} \iota(s').ub), \\ & min(\iota(s).ub, 1 - \sum_{s' \in S \wedge s' \neq s} \iota(s').lb)]. \end{aligned}$$

We are now ready to define the important notion of a weak instance. A weak instance describes the objects that can occur in a semistructured instance, the labels that can occur on the edges in an instance and constraints on the number of children an object might have. We will later define a probabilistic instance to be a weak instance with some probabilistic attributes.

Definition 3.8 *A* **weak instance** $\mathcal{W}$ *with respect to* $\mathcal{O}$*,* $\mathcal{L}$ *and* $\mathcal{T}$ *is a* 5*-tuple* $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$ *where:*

1. $V \subseteq \mathcal{O}$*.*
2. *For each object* $o \in V$ *and each label* $l \in \mathcal{L}$*,* $\mathsf{lch}(o, l)$ *specifies the set of objects that* **may** *be children of* o *with label* l*.*
3. τ *associates a type in* $\mathcal{T}$ *with each leaf vertex.*
4. val *associates a value in* $dom(\tau(o))$ *with each leaf object* o*.*
5. card *is mapping which constrains the number of children with a given label* l*.* card *associates with each object* $o \in V$ *and each label* $l \in \mathcal{L}$*, an integer-valued interval function,* $\mathsf{card}(o, l) = [min, max]$*, where* $min \geq 0$*, and* $max \geq min$*. We use* $\mathsf{card}(object, l).min$ *and* $\mathsf{card}(object, l).max$ *to refer to the lower and upper bounds respectively.*

A weak instance implicitly defines - for each object and each label - a set of potential sets of children. To see why, consider the following example.

Example 3.3 *Consider a weak instance with* $V = \{convoy, truck1, truck2, tank1, tank2\}$*. We may have* $\mathsf{lch}(convoy, truck) = \{truck1, truck2\}$ *indicating that truck1 and truck2 are possible truck-children of convoy. Likewise, we may have* $\mathsf{lch}(convoy, tank) = \{tank1, tank2\}$*. However, if* $\mathsf{card}(convoy, truck) = [0, 1]$*, then convoy can have at most one truck child. The set of sets of* truck*-children of convoy is thus* $\{\{\}, \{truck1\}, \{truck2\}\}$*. Likewise, if* $\mathsf{card}(convoy, tank) = [0, 2]$ *then convoy can have between zero and two (inclusive)* tank*-children. Thus there are twelve possiblities for the children of convoy.*

We formalize the reasoning in the above example below.

Definition 3.9 *Suppose* $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$ *is a weak instance and* $o \in V$ *and* l *is a label. A set* c *of objects in* V *is a* **potential** l*-child set of* o *w.r.t. the above weak instance iff:*

1. *If* $o' \in \mathsf{c}$ *then* $o' \in \mathsf{lch}(o, l)$ *and*

2. *The cardinality of* c *lies in the closed interval* $\mathsf{card}(o, l)$.

We us the notation $\mathsf{PL}(o, l)$ *to denote the set of all potential* l*-child sets of* o*.*

We are now in a position to define the potential children of an object o.

Definition 3.10 *Suppose* $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$ *is a weak instance and* $o \in V$*. A* **potential child set** *of* o *is any set* Q *of subsets of* V *such that* $Q = \bigcup H$ *where* H *is a hitting set of* $\{\mathsf{PL}(o, l) \mid (\exists o')o' \in \mathsf{lch}(o, l)\}$.

We use $\mathsf{PC}(o)$ *to denote the set of all potential child sets of* o *w.r.t. a weak instance.*

It is important to note that once a weak instance is fixed, $\mathsf{PC}(o)$ is well defined for each o. We will use this in our definition of a probabilistic instance.

Definition 3.11 *A* **probabilistic instance** $\mathcal{I}$ *is a* 6*-tuple* $\mathcal{I} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$ *where:*

1. $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$ *is a weak instance and*
2. ipf *is a mapping which associates with each non-leaf object* $o \in V$*, an interval probability function* ipf *w.r.t.* $\mathsf{PC}(o)$*, where* $\mathsf{c} \in \mathsf{PC}(o)$ *and* $\mathsf{ipf}(o, \mathsf{c}) = [lb, ub]$.

Intuitively, a probabilistic instance consists of a weak instance, together with probability intervals associated with each potential child of each object in the weak instance. The following example builds upon preceding examples to illustrate the concept of a probabilistic instance.

o	l	$\mathsf{lch}(o, l)$
convoy	truck	{ truck1, truck2 }
convoy	tank	{ tank1, tank2 }
truck1	truck	{ }
truck1	tank	{ }
truck2	truck	{ }
truck2	tank	{ }
tank1	truck	{ }
tank1	tank	{ }
tank2	truck	{ }
tank2	tank	{ }

o	$\tau(o)$	$\mathsf{val}(o)$
truck1	truck-type	rover
truck2	truck-type	rover
tank1	tank-type	T-80
tank2	tank-type	T-72

o	l	$\mathsf{card}(o, l)$
convoy	truck	[0,1]
convoy	tank	[0,2]

$\mathsf{c} \in \mathsf{PC}(convoy)$	$\mathsf{ipf}(convoy, \mathsf{c})$
{ }	[0.02, 0.05]
{ truck1}	[0.05, 0.08]
{ truck2}	[0.05, 0.08]
{ tank1}	[0.07, 0.1]
{ tank2}	[0.07, 0.1]
{ tank1, tank2}	[0.08, 0.15]
{ truck1, tank1}	[0.1, 0.2]
{ truck1, tank2}	[0.1, 0.2]
{ truck2, tank1}	[0.1, 0.2]
{ truck2, tank2}	[0.1, 0.2]
{ truck1, tank1, tank2}	[0.09, 0.18]
{ truck2, tank1, tank2}	[0.09, 0.18]

Figure 2: A probabilistic instance for the surveillance domain.

Example 3.4 *Figure 2 shows a very simple probabilistic instance. The set* $\mathcal{O}$ *of objects is* $\{convoy, truck1, truck2, tank1, tank2\}$*. As shown in the figure, the objects that can be* $truck$*-children of* $convoy$ *are:* $\mathsf{lch}(convoy, truck) = \{truck1, truck2\}$*. The objects that can be* $tank$*-children of* $convoy$ *are:* $\mathsf{lch}(convoy, tank) = \{tank1, tank2\}$*. The cardinality constraints for object convoy say that it can have 0 to 1 truck-children and 0 to 2 tank-children. The table on the right of Figure 2 shows the* ipf *of each potential child of convoy. Intuitively,* $\mathsf{ipf}(\{\}) = [0.02, 0.05]$ *says that the probability of having no truck and no tank in the convoy is between* 0.02 *and* 0.05*.*

The components $\mathcal{O}, \mathcal{L}, \mathcal{T}$ of a probabilistic instance are identical to those in a semistructured instance. However, in a probabilistic instance, there is uncertainty over:

- The number of sub-objects of an object;
- The identity of the sub-objects.

This uncertainty is captured through the function ipf. ipf may be defined extensionally, defining an interval probability for *each* potential child of every object. Or we may define ipf more compactly, in the case where there are some symmetries or independence constraints that can be exploited in the representation. For example, if the occurrence of each category of labelled objects is independent, then we can simply specify interval constraints for each subset of objects with the same label. In our example above, the tanks and trucks are indistinguishable so the ipf for say $\{truck1, tank1\}$ is the same as for $\{truck1, tank2\}$.

It is important to note that at any given point in time, the world is in some state. Uncertainty arises because we do not know what that state is. When states of the world are represented by semistructured instances, we can represent our uncertainty as a distribution over possible semistructured instances. A probabilistic instance *implicitly* is shorthand for a set of (possible) semistructured instances - these are the only instances that are *compatible* with the information we do have about the actual world state. This important notion of compatibility is given below.

Definition 3.12 *Let $\mathcal{S} = (V, E, \ell, \tau_{\mathcal{S}}, \mathsf{val}_{\mathcal{S}})$ be a semistructured instance over a set of objects $\mathcal{O}$, a set of labels $\mathcal{L}$ and a set of types $\mathcal{T}$ and let $\mathcal{W} = (V, \mathsf{lch}_{\mathcal{W}}, \tau_{\mathcal{W}}, \mathsf{val}_{\mathcal{W}}, \mathsf{card})$ be a weak instance. $\mathcal{S}$ is compatible with $\mathcal{W}$ if for each o in V:*

- *If o is a leaf in $\mathcal{S}$, then $\tau_{\mathcal{S}}(o) = \tau_{\mathcal{W}}(o)$ and $\mathsf{val}_{\mathcal{S}}(o) = \mathsf{val}_{\mathcal{W}}(o)$.*
- *If o is not a leaf in $\mathcal{S}$ then*
  - *For each edge (o, o') with label l in $\mathcal{S}$, $o' \in \mathsf{lch}_{\mathcal{W}}(o, l)$,*
  - *For each label $l \in \mathcal{L}$,*
    $$card(o, l).min \leq k \leq card(o, l).max$$
    where $k = |\{o' | (o, o') \in E \wedge \ell(E) = l\}|$, and
  - *$\mathsf{C}_{\mathcal{S}}(o) = \{o' | (o, o') \in E\}$, i.e., the set of children of o in instance $\mathcal{S}$.*

We use $\mathcal{D}(\mathcal{W})$ to denote the set of all semistructured instances that are compatible with a weak instance $\mathcal{W}$. Similarly, for a probabilistic instance $\mathcal{I} = (V, \mathsf{lch}_{\mathcal{I}}, \tau_{\mathcal{I}}, \mathsf{val}_{\mathcal{I}}, \mathsf{card}, \mathsf{ipf})$ we use $\mathcal{D}(\mathcal{I})$ to denote the set of all semistructured instances that are compatible with $\mathcal{I}$'s associated weak instance $\mathcal{W} = (V, \mathsf{lch}_{\mathcal{I}}, \tau_{\mathcal{I}}, \mathsf{val}_{\mathcal{I}}, \mathsf{card})$.

4 Semantics

In this section, we develop the semantics for probabilistic semistructured databases. We will introduce the global and local interpretations and then define the satisfaction of a probabilistic instance.

4.1 Global and Local Interpretations

In this section, we develop the semantics for a given global or local interpretation. We start by defining the important concept of a *global interpretation*.

Definition 4.1 *Suppose we have a weak instance $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$. A* **global interpretation** *$\mathcal{P}$ is a mapping from $\mathcal{D}(\mathcal{W})$ to $[0, 1]$ such that $\Sigma_{S \in \mathcal{D}(\mathcal{W})} \mathcal{P}(S) = 1$.*

Intuitively, a global interpretation is a distribution over the semistructured instances compatible with a weak instance.

We will show how the global semantics are defined in terms of local semantics. We first introduce the notion of a local probability model for a set of children of an object.

Definition 4.2 *Suppose $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$ is a weak instance. Let $o \in V$ be a non-leaf object. An* **object probability function** *(OPF for short)for o w.r.t. $\mathcal{W}$ is a mapping $\omega : \mathsf{PC}(o) \rightarrow [0, 1]$ such that*

$$\Sigma_{\mathsf{c} \in \mathsf{PC}(o)} \omega(\mathsf{c}) = 1.$$

An object probability function provides the model theory needed to study a single non-leaf object (and its children) in a probabilistic instance.

Definition 4.3 *Suppose $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$ is a weak instance. A* **local interpretation** *is a mapping $\wp$ from the set of non-leaf objects $o \in V$ to object probability functions, i.e. $\wp(o)$ returns an OPF for o w.r.t. $\mathcal{W}$.*

Intuitively, a local interpretation specifies, for each non-leaf object in the weak instance, an object probability function.

Example 4.1 (example probability interpretation) *A possible local interpretation $\wp$ that satisfies the example instance in Example 3.4 is as follows. $\wp(convoy) = \omega$ where $\omega(\{\}) = 0.02$, $\omega(\{truck1\}) = 0.05$, $\omega(\{truck2\}) = 0.05$, $\omega(\{tank1\}) = 0.1$, $\omega(\{tank2\}) = 0.1$, $\omega(\{tank1, tank2\}) = 0.08$, $\omega(\{ truck1, tank1 \}) = 0.1$, $\omega(\{ truck1, tank2 \}) = 0.1$, $\omega(\{ truck2, tank1 \}) = 0.1$, $\omega(\{ truck2, tank2 \}) = 0.1$, $\omega(\{ truck1, tank1, tank2 \}) = 0.1$, $\omega(\{ truck2, tank1, tank2 \}) = 0.1$.*

The local interpretation defines the local semantics for an object. In addition, it enables us to define the global semantics for our model. The probability of any particular instance is just the product of the OPF entries corresponding to each object in the instance and its children. We are now going to define two operators for describing the relationship between the local interpretation and the global interpretation.

Definition 4.4 ($\tilde{W}$ operator) *Let $\wp$ be local interpretation for a weak instance $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$. Then $\tilde{W}(\wp)$ returns a function defined as follows: for any instance $S \in \mathcal{D}(\mathcal{W})$*

$$\tilde{W}(\wp)(S) = \prod_{o \in S} \wp(o)(\mathsf{C}_S(o)).$$

In order to use this definition for the semantics of our model, we must show that the above function is in fact a legal global interpretation.

Theorem 2 *Suppose $\wp$ is a local interpretation for a weak instance $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$. Then $\tilde{W}(\wp)$ is a global interpretation for $\mathcal{W}$.*

Proof:

The proof is by induction.

The length of the path from the root r of a rooted directed acyclic graph G to an object o is the depth of o in G. The largest depth of any object in G is the height of G. We will call the root of a graph of height k as o_k. Now, we define k sets, namely $O_{G,k}, O_{G,k-1}, \ldots, O_{G,0}$, which contain objects of depth $0, 1, \ldots, k$.[2]

[2]We intentionally make the subscript of O as opposite to the depth it is corresponding so that the remaining parts of the proof is simpler and easier to understand when we are using induction.

A set $O_{G,k}$ is defined to contain o_k only. $O_{G,j-1}$ is defined as the union of the sets of children of objects in $O_{G,j}$, minus $O_{G,j} \cup \ldots \cup O_{G,k}$, i.e.,

$$O_{G,j-1} = \bigcup_{o \in O_{G,j}} \mathsf{C}(o) - \bigcup_{m=j}^{k} O_{G,m}.$$

Intuitively, the depth of objects in $O_{G,j}$ is $k - j$.

Suppose $\wp(o)$ returns an OPF ω_o for o w.r.t. $\mathcal{W}$.

Consider the case that the height of $\mathcal{W}$ is 1. The root o_1 is the only object in any $S \in \mathcal{D}(\mathcal{W})$ that can have children. Thus,

$$\begin{aligned} \tilde{W}(\wp)(S) &= \prod_{o \in S} \wp(o)(\mathsf{C}_S(o)) \\ &= \wp(o_1)(\mathsf{C}_S(o_1)) = \omega_{o_1}(\mathsf{C}_S(o_1)) \end{aligned}$$

In order for $\tilde{W}(\wp)$ to be a global interpretation for $\mathcal{W}$, the sum of $\tilde{W}(\wp)(S)$ over all S compatible with $\mathcal{W}$ should be equal to one. In this case, each distinct S has the object o_1 to contain a distinct potential child set. By Definition 4.2, the sum always gives one.

$$\begin{aligned} \sum_{S \in \mathcal{D}(\mathcal{W})} \tilde{W}(\wp)(S) &= \sum_{\mathsf{c}_0 \in \mathsf{PC}(o_1)} \omega_{o_1}(\mathsf{c}_0) \\ &= 1. \end{aligned}$$

Consider the case that the height of $\mathcal{W}$ is 2. The root is denoted as o_2.

$$\begin{aligned} & \tilde{W}(\wp)(S) \\ = & \prod_{o \in S} \wp(o)(\mathsf{C}_S(o)) \\ = & \wp(o_2)(\mathsf{C}_S(o_2)) \prod_{o_1 \in \mathsf{C}_S(o_2)} \wp(o_1)(\mathsf{C}_S(o_1)) \\ = & \omega_{o_2}(\mathsf{C}_S(o_2)) \prod_{o_1 \in \mathsf{C}_S(o_2)} \omega_{o_1}(\mathsf{C}_S(o_1)) \end{aligned}$$

Since $\mathcal{D}(\mathcal{W})$ contains all possible compatible instances and the set of all potential child sets of an object is independent of other objects, $\mathcal{D}(\mathcal{W})$ will then contains all possible combinations of every potential child set of every object.

$$\begin{aligned} & \sum_{S \in \mathcal{D}(\mathcal{W})} \tilde{W}(\wp)(S) \\ = & \sum_{\mathsf{c}_1 \in \mathsf{PC}(o_2)} \omega_{o_2}(\mathsf{c}_1) \prod_{o_1 \in \mathsf{c}_1} \sum_{\mathsf{c}_0 \in \mathsf{PC}(o_1)} \omega_{o_1}(\mathsf{c}_0) \\ = & \sum_{\mathsf{c}_1 \in \mathsf{PC}(o_2)} \omega_{o_2}(\mathsf{c}_1) \prod_{o_1 \in \mathsf{c}_1} 1 \\ = & \sum_{\mathsf{c}_1 \in \mathsf{PC}(o_2)} \omega_{o_2}(\mathsf{c}_1) \\ = & 1. \end{aligned}$$

Assume that the theorem is true for the cases that the height of $\mathcal{W}$ is $1, \ldots, k+1$. Now, let us consider the case that the height of $\mathcal{W}$ is $k+2$.

$$\begin{aligned}
& \tilde{W}(\wp)(S) \\
= & \prod_{o \in S} \wp(o)(\mathsf{C}_S(o)) \\
= & \prod_{j=1}^{k+2} \prod_{o_j \in O_{G,j}} \wp(o_j)(\mathsf{C}_S(o_j)) \\
= & \prod_{j=1}^{k+2} \prod_{o_j \in O_{G,j}} \omega_{o_j}(\mathsf{C}_S(o_j))
\end{aligned}$$

While summing up over all compatible instances, we can use the assumption that the subgraph (height $k+1$) of $\mathcal{W}$ without the root o_{k+2} has the product of sum equal to one.[3]

$$\begin{aligned}
& \sum_{S \in \mathcal{D}(\mathcal{W})} \tilde{W}(\wp)(S) \\
= & \sum_{\mathsf{c}_{k+1} \in \mathsf{PC}(o_{k+2})} \omega_{o_{k+2}}(\mathsf{c}_{k+1}) \\
& \times \prod_{j=1}^{k+1} \left(\prod_{o_j \in O_{\mathcal{W},j}} \left(\sum_{\mathsf{c}_{j-1} \in \mathsf{PC}(o_j)} \omega_{o_j}(\mathsf{c}_{j-1}) \right) \right) \\
= & \sum_{\mathsf{c}_{k+1} \in \mathsf{PC}(o_{k+2})} \omega_{o_{k+2}}(\mathsf{c}_{k+1}) \times 1 \\
= & \ 1.
\end{aligned}$$

∎

An important question is whether we can go the other way: from a global interpretation, can we find a local interpretation for a weak instance $\mathcal{W}(V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$? It turns out that we can **if** the global interpretation can be factored in a manner consistent with the structure constraints imposed by $\mathcal{W}(V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$.

One way to ensure that this is possible is to impose a set of independence constraints.

Definition 4.5 *Suppose $\mathcal{P}$ is a global interpretation and $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$ is a weak instance. $\mathcal{P}$* **satisfies** *$\mathcal{W}$ iff for every non-leaf object $o \in V$ and each $c \in \mathsf{PC}(o)$, it is the case that*[4]*:*

$$\mathcal{P}(c|o, \mathsf{ndes}(o)) = \mathcal{P}(c|o)$$

In other words, given that o occurs in the instance, the probability of any potential children c of o is independent of the nondescendants of o in the instance (this may not be quite precise enough– it is independent of any *possible* set of nondescendants). From now on, given a weak instance $\mathcal{W}$, we will only consider $\mathcal{P}$ that satisfies $\mathcal{W}$.

Furthermore, given a global interpretation that satisfies a weak instance, we can find a local interpretation associated with it in the following (informally described[5]) manner.

[3] Although now the subgraph mave have more than one root, it can be proved in a similar way that the product of sum equal to one.

[4] Here, $\mathcal{P}(c|o)$ is the probability of c being children of o given that o exists. The notation of $\mathcal{P}(c|o, A)$ means the probability of c being children of o given that o and A exists, where A is a set of objects.

[5] The formal description is omitted due to space constraints.

Definition 4.6 ($\tilde{D}$ operator) *Suppose* $\mathsf{c} \in \mathsf{PC}(o)$ *for some non-leaf object* o *and suppose* $\mathcal{P}$ *is a global interpretation.* $\omega_{\mathcal{P},o}$*, is defined as follows.*

$$\omega_{\mathcal{P},o}(\mathsf{c}) \quad = \quad \frac{\Sigma_{S \in \mathcal{D}(\mathcal{W}) \wedge o \in S \wedge \mathsf{C}_S(o) = \mathsf{c}} \mathcal{P}(S)}{\Sigma_{S \in \mathcal{D}(\mathcal{W}) \wedge o \in S} \mathcal{P}(S)}.$$

Then, $\tilde{D}(\mathcal{P})$ *returns a function defined as follows: for any non-leaf object* o*,* $\tilde{D}(\mathcal{P})(o) = \omega_{\mathcal{P},o}$*.*

Intuitively, we construct $\omega_{\mathcal{P},o}(\mathsf{c})$ as follows. Find all semistructured instances S that are compatible with $\mathcal{W}$ and eliminate those for which o's set of children is not c. The sum of the (normalized) probabilities assigned to the remaining semistructured instances by $\mathcal{P}$ is assigned to c by the OPF $\omega_{\mathcal{P},o}(\mathsf{c})$. By doing this for each non-leaf object o and each of its potential child sets, we get a local interpretation.

Theorem 3 *Suppose* $\mathcal{P}$ *is a global interpretation for a weak instance* $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$*. Then* $\tilde{D}(\mathcal{P})$ *is a local interpretation for* $\mathcal{W}$*.*

Proof: From Definition 4.6, $\tilde{D}(\mathcal{P})(o) = \omega_{\mathcal{P},o}$ is an OPF for o because $\Sigma_{\mathsf{c}} \omega_{\mathcal{P},o}(\mathsf{c}) = 1$. By Definition 4.3, $\tilde{D}(\mathcal{P})$ is a local interpretation because for every non-leaf object o, $\tilde{D}(\mathcal{P})(o)$ returns an OPF for o. ∎

Theorem 4 *Suppose* $\wp$ *is a local interpretation for a weak instance* $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$*. Then,* $\tilde{D}(\tilde{W}(\wp)) = \wp$*.*

Proof:

Here we try to abuse some notations for convenience. Given a graph G and a subset A of objects in G, $G - A$ is a subgraph of G after removing the set A of objects and all edges connected to A. Define $G_{ndes(o)} = G - o - \mathsf{des}_G(o)$ for any acyclic directed graph G, i.e., $G_{ndes(o)}$ is a subgraph of G without an object o and its descendants. Define $G_{des(o)} = G - o - \mathsf{ndes}_G(o)$ for any acyclic directed graph G, i.e., $G_{des(o)}$ is a subgraph of G containing descendants of an object o only. In this proof, we will treat instances S and weak instances $\mathcal{W}$ as graphs and the above notations will be used on them.

Define $\mathcal{D}_{ndes(o)}$ as function of $\mathcal{W}$ which returns a set of "compatible" subgraphs of $\mathcal{W}$ only containing nondescendants of an object o, i.e., $\mathcal{D}_{ndes(o)}(\mathcal{W}) = \{S - o - \mathsf{des}_{\mathcal{W}}(o) | S \in \mathcal{D}(\mathcal{W})\}$ where $\mathsf{des}_{\mathcal{W}}(o)$ are the descendants of o in $\mathcal{W}$. Similarly, define $\mathcal{D}_{des(o)}$ as function of $\mathcal{W}$ which returns a set of "compatible" subgraphs of $\mathcal{W}$ only containing an object o and its descendants, i.e., $\mathcal{D}_{des(o)}(\mathcal{W}) = \{S - \mathsf{ndes}_{\mathcal{W}}(o) | S \in \mathcal{D}(\mathcal{W})\}$ where $\mathsf{ndes}_{\mathcal{W}}(o)$ are the nondescendants of o in $\mathcal{W}$.

We need to prove that for any non-leaf object o and its any potential child set c, $\tilde{D}(\tilde{W}(\wp))(o)(\mathsf{c}) = \wp(o)(\mathsf{c})$. The formula in Definition 4.4 can be rewritten as the following:

$$\tilde{W}(\wp)(S) = \alpha_S \times \wp(o)(\mathsf{C}_S(o)) \times \beta_S$$

where

$$\alpha_S \quad = \quad \prod_{o' \in S \wedge o' \in \mathcal{W}_{ndes(o)}} \wp(o')(\mathsf{C}_S(o')),$$

$$\beta_S \quad = \quad \prod_{o' \in S \wedge o' \notin \mathcal{W}_{ndes(o)}} \wp(o')(\mathsf{C}_S(o')).$$

Intuitively, α_S is the product of the probabilities of children of all objects excluding o and its descendants. β_S is the product of the probabilities of children of all objects including only descendants of o. Here, descendants and nondescendants are those of the object o in $\mathcal{W}$.

It is obvious that for any two instances $S, S' \in \mathcal{D}(\mathcal{W})$, if $S_{ndes(o)} = S'_{ndes(o)}$, then $\alpha_S = \alpha_{S'}$. Similarly, if $S_{des(o)} = S'_{des(o)}$, then $\beta_S = \beta_{S'}$. For any $X \in \mathcal{D}_{ndes(o)}(\mathcal{W})$, we definea new term $\alpha_{(X)} = \alpha_S$ if $X = S_{ndes(o)}$. Similarly, for any $Y \in \mathcal{D}_{des(o)}(\mathcal{W})$, we define a new term $\beta_{(Y)} = \beta_S$ if $Y = S_{des(o)}$.

The formula in Definition 4.6 to compute $\omega_{\tilde{W}(\wp),o}(\mathsf{c})$ can be rewritten as:

$$\tilde{D}(\tilde{W}(\wp))(o)(\mathsf{c}) = \omega_{\tilde{W}(\wp),o}(\mathsf{c}) = \frac{a}{a+b}$$

where

$$\begin{aligned}
a &= \sum_{S \in \mathcal{D}(\mathcal{W}) \wedge o \in S \wedge \mathsf{C}_S(o) = \mathsf{c}} \alpha_S \times \wp(o)(\mathsf{C}_S(o)) \times \beta_S \\
&= \sum_{X \in \mathcal{D}_{ndes(o)}(\mathcal{W})} \alpha_{(X)} \\
&\quad \times \wp(o)(\mathsf{c}) \times \sum_{Y \in \mathcal{D}_{des(o)}(\mathcal{W})} \beta_{(Y)} \\
&\quad \text{(the reasoning in this step is similar to Theorem 2)} \\
&= \sum_{X \in \mathcal{D}_{ndes(o)}(\mathcal{W})} \alpha_{(X)} \times \wp(o)(\mathsf{c}) \times 1, \\
&\quad \text{(the last term can be proved similarly to Theorem 2)} \\
b &= \sum_{S \in \mathcal{D}(\mathcal{W}) \wedge o \in S \wedge \mathsf{C}_S(o) \neq \mathsf{c}} \alpha_S \times \wp(o)(\mathsf{C}_S(o)) \times \beta_S \\
&= \sum_{X \in \mathcal{D}_{ndes(o)}(\mathcal{W})} \alpha_{(X)} \\
&\quad \times \sum_{\mathsf{c}_i \in \mathsf{PC}(o) \text{ where } \mathsf{c}_i \neq \mathsf{c}} \wp(o)(c_i) \\
&\quad \times \sum_{Y \in \mathcal{D}_{des(o)}(\mathcal{W})} \beta_{(Y)} \\
&\quad \text{(more precisely, for the last term,} \\
&\quad \text{we require } \exists S \in \mathcal{D}(\mathcal{W}) \text{ such that} \\
&\quad S_{\mathsf{ndes}(o)} = X \text{ and } S_{\mathsf{des}(o)} = Y.\text{)} \\
&= \sum_{X \in \mathcal{D}_{ndes(o)}(\mathcal{W})} \alpha_{(X)} \\
&\quad \times \sum_{\mathsf{c}_i \in \mathsf{PC}(o) \text{ where } \mathsf{c}_i \neq \mathsf{c}} \wp(o)(\mathsf{C}_S(o)) \times 1 \\
&= \sum_{X \in \mathcal{D}_{ndes(o)}(\mathcal{W})} \alpha_{(X)} \times (1 - \wp(o)(\mathsf{c})).
\end{aligned}$$

Thus,

$$\omega_{\tilde{W}(\wp),o}(\mathsf{c}) = \frac{\wp(o)(\mathsf{c})}{\wp(o)(\mathsf{c}) + 1 - \wp(o)(\mathsf{c})} = \wp(o)(\mathsf{c}).$$

∎

Theorem 5 *Suppose $\mathcal{P}$ is a global interpretation for a weak instance $\mathcal{W} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card})$. Then, $\tilde{W}(\tilde{D}(\mathcal{P})) = \mathcal{P}$.*

Proof:

Because $\mathcal{P}$ satisfies $\mathcal{W}$, the probability of a child set of a given object is independent of other objects, and so we can factorize $\mathcal{P}(S)$ to the product of the conditional probabilities of child sets of all non-leaf objects. Furthermore, there exists an OPF ω_o for every non-leaf object o such that for every $S \in \mathcal{D}(\mathcal{W})$, $\mathcal{P}(S) = \prod_{o \in S} \omega_o(\mathsf{C}(o))$.

Now, we need to prove that for any non-leaf object o and its any potential child set c, $\tilde{D}(\mathcal{P})(o)(\mathsf{c}) = \omega_o(\mathsf{c})$, i.e.,

$$\begin{aligned} \tilde{D}(\mathcal{P})(o)(\mathsf{c}) &= \frac{\Sigma_{S \in \mathcal{D}(\mathcal{W}) \wedge o \in S \wedge \mathsf{C}_S(o) = \mathsf{c}} \mathcal{P}(S)}{\Sigma_{S \in \mathcal{D}(\mathcal{W}) \wedge o \in S} \mathcal{P}(S)} \\ &= \omega_o(\mathsf{c}). \end{aligned}$$

After that, then we can show that for every S,

$$\begin{aligned} \tilde{W}(\tilde{D}(\mathcal{P}))(S) &= \prod_{o \in S} \tilde{D}(\mathcal{P})(o)(\mathsf{C}(o)) \\ &= \prod_{o \in S} \omega_o(\mathsf{C}(o)) \\ &= \mathcal{P}(S). \end{aligned}$$

Now, the question is how to prove $\tilde{D}(\mathcal{P})(o)(\mathsf{c}) = \omega_o(\mathsf{c})$. We can define a local interpretation $\wp$ such that $\wp(o) = \omega_o$. Furthermore,

$$\begin{aligned} \tilde{W}(\wp)(S) &= \prod_{o \in S} \wp(o)(\mathsf{C}_S(o)) \\ &= \prod_{o \in S} \omega_o(\mathsf{C}(o)) \\ &= \mathcal{P}(S) \end{aligned}$$

Thus, $\tilde{W}(\wp) = \mathcal{P}$. As a result, now what we need to prove has become: for any non-leaf object o and its any potential child set c, $\tilde{D}(\tilde{W}(\wp))(o)(\mathsf{c}) = \wp(o)(\mathsf{c})$. This is exactly the same as that in the proof of Theorem 4. ∎

4.2 Satisfaction of probabilistic instance

Up until now, we have been discussing the semantics in terms of global interpretations and local interpretations, but we have not yet discussed probabilistic instances. We are now ready to address the following important problem: under what conditions does a local interpretation satisfy a probabilistic instance, and under what conditions does a global interpretation satisfy a probabilistic instance. To do so, we will first define what it means for an OPF and a local interpretation to satisfy a non-leaf object. The definitions of when a local interpretation and a global interpretation satisfy a probabilistic instance will then be straightforward.

A probabilistic instance puts constraints on the probability specifications for objects. We associate a set of object constraints with each non-leaf object as follows.

Definition 4.7 (object constraints) *Suppose $\mathcal{I} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$ is a probabilistic instance, and $o \in V$ is a non-leaf node. We associate with o, a set of constraints called* object constraints*, denoted $\mathsf{OC}(o)$, as follows. For each $\mathsf{c} \in \mathsf{PC}(o)$, $\mathsf{OC}(o)$ contains the constraint*

$$\mathsf{ipf}(o, \mathsf{c}).lb \leq p(\mathsf{c}) \leq \mathsf{ipf}(o, \mathsf{c}).ub$$

where $p(\mathsf{c})$ is a real-valued variable denoting the probability that c is the actual set of children of o. $\mathsf{OC}(o)$ also includes the following constraint

$$\Sigma_{\mathsf{c} \in \mathsf{PC}(o)} p(\mathsf{c}) = 1.$$

Example 4.2 *Consider the probabilistic instance defined in Example 3.4. $\mathsf{OC}(convoy)$ is defined as follows:*

- $0.02 \leq p(\{\}) \leq 0.05$
- $0.05 \leq p(\{truck1\}) \leq 0.08$
- $0.05 \leq p(\{truck2\}) \leq 0.08$
- $0.07 \leq p(\{tank1\}) \leq 0.1$
- $0.07 \leq p(\{tank2\}) \leq 0.1$
- $0.08 \leq p(\{tank1, tank2\}) \leq 0.15$
- $0.1 \leq p(\{\ truck1, tank1\ \}) \leq 0.2$
- $0.1 \leq p(\{\ truck1, tank2\ \}) \leq 0.2$
- $0.1 \leq p(\{\ truck2, tank1\ \}) \leq 0.2$
- $0.1 \leq p(\{\ truck2, tank2\ \}) \leq 0.2$
- $0.09 \leq p(\{\ truck1, tank1, tank2\ \}) \leq 0.18$
- $0.09 \leq p(\{\ truck2, tank1, tank2\ \}) \leq 0.18$

Intuitively, an OPF satisfies a non-leaf object iff the assignment made to the potential children by the OPF is a solution to the constraints associated with that object. Obviously, a probability distribution w.r.t. $\mathsf{PC}(o)$ over ipf is a solution to $\mathsf{OC}(o)$.

Definition 4.8 (object satisfaction) *Suppose $\mathcal{I} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$ is a probabilistic instance, $o \in V$ is a non-leaf node, ω is an OPF for o, and $\wp$ is a local interpretation. ω satisfies o iff ω is a probability distribution w.r.t. $\mathsf{PC}(o)$ over ipf. $\wp$ satisfies o iff $\wp(o)$ satisfies o.*

Example 4.3 *Consider the probabilistic instance defined in Example 3.4, the probability interpretation defined in Example 4.1 and the $\mathsf{OC}(convoy)$ defined in Example 4.2. Since the assignment made to the potential children of $convoy$ by the OPF ω is a solution to the constraints $\mathsf{OC}(convoy)$ associated with $convoy$, ω is a probability distribution w.r.t. $\mathsf{PC}(convoy)$ over ipf. Thus, ω satisfies $convoy$ and the local interpretation $\wp$ satisfies $convoy$.*

We are now ready to extend the above definition to the case of satisfaction of a probabilistic instance by a local interpretation.

Definition 4.9 (local satisfaction of a prob. inst.) *Suppose $\mathcal{I} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$ is a probabilistic instance, and $\wp$ is a local interpretation. $\wp$ satisfies $\mathcal{I}$ iff for every non-leaf object $o \in V$, $\wp(o)$ satisfies o.*

Example 4.4 *Consider the probabilistic instance defined in Example 3.4, the local interpretation $\wp$ defined in Example 4.1 and the satisfaction of $convoy$ by $\wp$ in Example 4.3. Since $convoy$ is the only non-leaf object, $\wp$ satisfies the example probabilistic instance.*

Similarly, a global interpretation $\mathcal{P}$ satisfies a probabilistic instance if the OPF computed by using $\mathcal{P}$ can satisfy the object constraints of each non-leaf object.

Definition 4.10 (global satisfaction of a prob. inst.) *Suppose $\mathcal{I} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$ is a probabilistic instance, and $\mathcal{P}$ is a global interpretation. $\mathcal{P}$ satisfies $\mathcal{I}$ iff for every non-leaf object $o \in V$, $\tilde{D}(\mathcal{P})(o)$ satisfies o, i.e., $\tilde{D}(\mathcal{P})$ satisfies $\mathcal{I}$.*

Corollary 1 (equivalence of local and global satisfaction) *Suppose $\mathcal{I} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$ is a probabilistic instance, and $\wp$ is a local interpretation. Then $\wp$ satisfies $\mathcal{I}$ iff $\tilde{W}(\wp)$ satisfies $\mathcal{I}$.*

Proof: By Definition 4.10, $\tilde{W}(\wp)$ satisfies $\mathcal{I}$ iff $\tilde{D}(\tilde{W}(\wp))$ satisfies $\mathcal{I}$. By Theorem 4, $\tilde{D}(\tilde{W}(\wp)) = \wp$. Thus, it is trival that the corollary is true. ∎

The definition of consistency below is the usual one, requiring that at least one probabilistic interpretation satisfies the probabilistic instance in question.

Definition 4.11 (local consistency) *A probabilistic instance is* locally consistent *iff there is a local interpretation that satisfies it.*

Notice that in order to check consistency, we merely need to check, for every non-leaf object, whether there exists an OPF that satisfies it, i.e., whether there exists a probability distribution w.r.t. $\mathsf{PC}(o)$ over ipf w.r.t. $\mathsf{PC}(o)$.

Definition 4.12 (global consistency) *A probabilistic instance is* globally consistent *iff there is a global interpretation that satisfies it.*

By Lemma 1, we have the following theorem.

Theorem 6 (consistency of prob. inst.) *Every probabilistic instance is (locally and globally) consistent.*

Proof: By Definition 4.10 and Corollary 1, a probabilistic instance is globally consistent iff it is locally consistent. Thus, we only need to prove that every probabilistic instance is locally consistent. Suppose $\mathcal{I} = (V, \mathsf{lch}, \tau, \mathsf{val}, \mathsf{card}, \mathsf{ipf})$ is a probabilistic instance. For every non-leaf object $o \in V$, $\mathsf{OC}(o)$ are exactly the same constraints of the definition (in Definition 3.5) of a probability distribution w.r.t. $\mathsf{PC}(o)$ over $\mathsf{ipf}(o, \mathsf{c})$. By Lemma 1, there exists such a probability distribution P, so we can define an OPD ω_o for o such that $\forall \mathsf{c} \in \mathsf{PC}(o)$, $\omega_o(\mathsf{c}) = P(\mathsf{c})$. ω_o is a probability distribution w.r.t. $\mathsf{PC}(o)$ over ipf w.r.t. $\mathsf{PC}(o)$, so it satisfies o. Thus, for each non-leaf object $o \in V$, we can define an OPD ω_o that satisfies o. Then we can define a local interpretation $\wp$ such that for every non-leaf object $o \in V$, $\wp(o) = \omega_o$. Therefore, for every non-leaf object $o \in V$, $\wp(o)$ satisfies o, so $\wp$ satisfies $\mathcal{I}$. ∎

5 Related Work

There have been work done on storing probabilistic information in relational databases[7], temporal databases[4], object databases[5], etc.

The work of Dekhtyar et al.[3] seems to be similar to ours, but in fact theirs and ours are on different things. They tried to introduce a semistructured model to support storage and querying of probabilistic information in flexible forms such as a simple interval probability distribution, a joint interval probability distribution, or a simple or joint conditional interval probability distribution. Their model allows to use an object (semistructured probabilistic object or SPO) to represent the probability table of one or more random variable, the extended context and the extended conditionals. An SPO itself can be represented in a semistructured way, but its main body is just a flat table. It cannot show the semistructured relationship among variables. In contrast, our model is based on the widely used model OEM[8], which allows data to be represented in a real semistructured manner. We modify the syntax and semantics of the model by introducing cardinality and interval probability function to demonstrate the uncertainty of the number and the identity of objects existing in possible worlds. Every possible world is a semistructured instance compatible to the probabilistic instance. The representation of a possible world (semistructured instance) is the same as the one widely accepted nowadays. However, the model of Dekhtyar et al. cannot do that. Theirs also requires random variables to have distinct variable names (in our model, they are the children connected to their parents with the same edge label). Consequently, theirs cannot allow two or more variables with he same variable names (no matter their values are the same or different) in a single possible world. Theirs also cannot capture the uncertainty of cardinality. On the other hand, our model can represent their table by, for each set of random variables with the same variable names, defining a set of children (with the possible variable values) connected to their parent with the same edge label (set as the variable name). The cardinality associates with the parent object with each label is set to $[1, 1]$ so that each random variable can have exactly one value in each possible world. The extended context and extended conditionals in SPO can be represented by two subtrees with corresponding edge labels and values connected to the parent object.

We are going to develop an algebra for our model. SAL [2] and TAX [6] are two algebras for semistructured data. However, we will not use them directly due to the following reasons. SAL binds objects to variables, manipulates the bindings and then removes bindings constructing a result. Our algebra is required to manipulate the graph structure of semistructured data directly because if the parent-child relationship among objects are lost, then there will be problems in dealing with ipf. TAX uses a pattern tree to extract subsets of nodes (called witness trees), one for each embedding of the pattern tree in an input tree (instance). Its algebraic operations are similar to what we want in some aspects. The reason that we cannot use theirs directly is the fixed structure of the result, e.g., fixed number of children, which reduces the uncertainty of cardinality.

6 Conclusions

In this paper, we have identified the importance of a semistructured model with the ability to handle uncertainty. With a motivating example, we described the syntax as well as the semantics of our ProbSem probabilistic semistructured model. We have defined the interpretation and satisfaction of a probabilistic instance globally and locally. We have proved that the two interpretations are equivalent, which helps us to prove that the two satisfactions are also equivalent. Finally, we show that every probabilistic instance is locally and globally consistent. The ProbSem model has been built and we are now working on the development of an algebra for that and the implementation of a system based on that.

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web*. Morgan Kaufmann, 2000.

[2] C. Beeri and Y. Tzaban. Sal: An algebra for semistructured data and xml. In *Informal Proceedings of the ACM International Workshop on the Web and Databases (WebDB'99)*, Philadelphia, Pennsylvania, USA, June 1999.

[3] A. Dekhtyar, J. Goldsmith, and S.R. Hawkes. Semistructured models for interval probabilities. *Submitted paper*.

[4] A. Dekhtyar, R. Ross, , and V.S. Subrahmanian. Probabilistic temporal databases, i. *ACM Transactions on Database Systems*, 26(1), 2001.

[5] T. Eiter, J. Lu, T. Lukasiewicz, and V.S. Subrahmanian. Probabilistic object bases. *ACM Transactions on Database Systems*, September 2001.

[6] H.V. Jagadish, Laks V.S. Lakshmanan, and Divesh Srivastava. Tax: A tree algebra for xml. In *Proc. of Int. Workshop on Database Programming Languages (DBPL'01)*, Roma, Italy, September 2001.

[7] V.S. Lakshmanan, N. Leone, R. Ross, and V.S. Subrahmanian. Probview: A flexible probabilistic database system. *ACM Transactions on Database Systems*, 22(3):419–469, 1997.

[8] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. In *Proc. of the Eleventh International Conference on Data Engineering*, pages 251–260, Taipei, Taiwan, March 1995.