On the Expressiveness of Probabilistic XML Models

Serge Abiteboul, Benny Kimelfeld, Y. Sagiv, Pierre Senellart

VLDB Journal manuscript No.
(will be inserted by the editor)

On the Expressiveness of Probabilistic XML Models

Serge Abiteboul · Benny Kimelfeld · Yehoshua Sagiv · Pierre Senellart

Abstract Various known models of probabilistic XML can be represented as instantiations of the abstract notion of *p-documents*. In addition to ordinary nodes, p-documents have *distributional* nodes that specify the possible worlds and their probabilistic distribution. Particular families of p-documents are determined by the types of distributional nodes that can be used as well as by the structural constraints on the placement of those nodes in a p-document. Some of the resulting families provide natural extensions and combinations of previously studied probabilistic XML models.

The focus of the paper is on the expressive power of families of p-documents. In particular, two main issues are studied. The first is the ability to (efficiently) *translate* a given p-document of one family into another family. The second is *closure under updates,* namely, the ability to (efficiently) represent the result of updating the instances of a p-document of a given family as another p-document of that family. For both issues, we distinguish two variants corresponding to *value-based* and *object-based* semantics of p-documents.

Keywords XML · Probabilistic databases · Probabilistic XML · Expressiveness · Updates

1 Introduction

Many automatic tasks, particularly on the Web, generate uncertain data. Examples of these tasks include information extraction, natural-language processing and data mining. Moreover, in many of these tasks, information is described in a semistructured model, because representation by means of a hierarchy is natural, especially when the source (e.g., XML or HTML) is already in this form. Uncertain hierarchical information can be formalized in terms of a probabilistic XML space, that is, a probability distribution over a set of ordinary XML documents. A number of probabilistic XML models [2–8] have been proposed for facilitating a succinct description of those spaces. In addition to the models themselves, various problems of managing probabilistic XML data have been studied, such as query evaluation [3, 7, 9, 10], algebraic manipulation [4] and updates [2, 7].

For developing a system that manages probabilistic XML, a proper data model should be chosen; to do that, two questions have to be addressed. First, what kind of information is it desired to represent (e.g., how do different uncertain data items correlate)? Second, which management tasks does the system need to perform? As

Some of the results described in this paper were reported in [1, 2].

The work of Abiteboul and Senellart was supported by the Agence Nationale de la Recherche under grant Docflow O6-MDCA-005, and by the Webdam Grant of the European Research Council.

Some of the work of Benny Kimelfeld was done while he was at The Hebrew University.

The work of Kimelfeld and Sagiv was supported by The Israel Science Foundation (Grant 893/05).

Serge Abiteboul
INRIA Saclay – Île-de-France & Université Paris-Sud
Orsay, France
E-mail: serge.abiteboul@inria.fr

Benny Kimelfeld
IBM Almaden Research Center
E-mail: kimelfeld@us.ibm.com

Yehoshua Sagiv
Hebrew University of Jerusalem
Jerusalem, Israel
E-mail: sagiv@cs.huji.ac.il

Pierre Senellart
Institut Télécom; Télécom ParisTech; CNRS LTCI
Paris, France
E-mail: pierre.senellart@telecom-paristech.fr

a concrete example, van Keulen et al. [6] use a specific model to represent the result of integrating two XML documents (where uncertainty essentially follows from heuristics for entity resolution). One may want to use the model of [6] for representing similar data, but then, will one be able to (efficiently) realize the algebra of [4] or evaluate twig queries using the algorithm of [9]? The answer is not obvious, given the differences among the three data models.

A simple way of bridging the different models and techniques is to devise *translations* between the models. That is, given a probabilistic XML document represented in one model, we translate it into another model, and then manage the result using techniques devised for the latter model. Moreover, for this process to be practical, the translation should be *efficient.* As we later show, it may be the case that a translation between two specific models exists, but it necessarily entails a major blowup in the size of the data. Thus, understanding the ability to efficiently translate between the different models, which is a goal set by this work, has a central role in choosing the suitable model for a system and analyzing the implications of a specific choice. Moreover, if one already has an implemented system based on a specific model and yet wishes to use it for data of a different model, then translations are essentially the only way to go.

Another important property of a probabilistic XML model is the ability to represent interesting evolution of the probabilistic data. So, in addition to comparing the expressive power of probabilistic XML models (i.e., in terms of efficient translations), we study the ability of the models to handle *updates*. More particularly, we consider insertion and deletion of data items that are done at elements specified by queries. Conceptually, these operations are done on the possible worlds. We investigate the ability to apply these updates directly to a probabilistic XML document and the cost thereof. We do it in the context of specific models.

We begin with presenting a unified view of these different models in terms of *p-documents* that are trees with two types of nodes: ordinary and *distributional*. A p-document can be thought of as a probabilistic process that generates a random XML document in a conceptually simple way. Namely, each distributional node v chooses a subset of its children.[1] Therefore, each distributional node of a p-document should specify the probability distribution of choosing a subset of its children in the above random process. There are several types of distributional nodes that differ from one another in how they specify probabilities and in certain properties thereof.

We consider five types of distributional nodes: det for *deterministic*[2] (each child is chosen with probability 1); ind for *independent* (the choices of distinct children are independent); mux for *mutually exclusive* (at most one child can be chosen); exp for *explicit* (the probability of choosing each subset of children is explicitly given unless it is zero); and cie for *conjunction of independent events* (each child is chosen according to a conjunction of probabilistically independent events, which can be used globally throughout the p-document).

We define different families of p-documents in terms of the types of distributional nodes that are allowed. PrXML^{C}, where $C \subseteq \{\mathsf{ind}, \mathsf{mux}, \mathsf{det}, \mathsf{exp}, \mathsf{cie}\}$, denotes the family of p-documents that use the types appearing in the subset C. We also consider additional families by imposing the restriction that there are no hierarchies consisting entirely of distributional nodes (i.e., a distributional node cannot have a distributional child). $\mathsf{PrXML}^{C}_{|\not\mathcal{H}}$ denotes the family of p-documents that use the types of C and have no distributional hierarchies. We later show that practically all the probabilistic XML models that have been proposed in the literature can be defined in this way.

We thoroughly investigate the expressive power of the above families. To define expressiveness properly, one should realize that we are not interested in a p-document per se, but rather in the probability space (over XML documents) that it describes. Thus, two p-documents are *equivalent* if they define the same probability space (called *px-space* in this paper). Consequently, a family $\mathscr{F}_2$ is *(at least) as expressive as* $\mathscr{F}_1$ if for every p-document of $\mathscr{F}_1$, there is an equivalent p-document of $\mathscr{F}_2$ (if the opposite direction does not hold, then $\mathscr{F}_2$ is *more expressive*). Practically, however, $\mathscr{F}_2$ subsumes $\mathscr{F}_1$ only if we can find an efficient algorithm, such that given a p-document of $\mathscr{F}_1$, it computes an equivalent p-document of $\mathscr{F}_2$. So, our emphasis is on *efficient translators* between families of p-documents. Furthermore, we consider two types of translators: *o-translators* and *v-translators*. The former is based on the object semantics (i.e., two p-documents are equivalent if they describe identical px-spaces), whereas the latter subscribes to the value semantics (that is, equivalence means isomorphic px-spaces).

Figure 5.7 summarizes our results about efficient o-translations, which are obtained in Section 5. This figure is complete in the sense that if there is no directed path from a family $\mathscr{F}_1$ to a family $\mathscr{F}_2$, then there is no

[1] This is an oversimplification—see Section 3.1 for the precise details.

[2] It may seem that using det nodes is redundant, but actually they increase the expressive power when used together with some of the other types.

efficient o-translation from $\mathscr{F}_1$ to $\mathscr{F}_2$. Note that if there is an efficient o-translation, then there is also an efficient v-translation (but the converse is not necessarily true). We show that in many cases, if there is no efficient o-translation (between two specific families $\mathscr{F}_1$ and $\mathscr{F}_2$ of our framework), then there is no efficient v-translation as well. However, the existence of efficient v-translations (in the absence of efficient o-translations) remains an open problem in some cases. Thus, Figure 5.7 is not complete with respect to v-translations, but it gives a fairly good picture.

We partially deal with the above open problems of Section 5 in Section 6, where we consider p-documents with the restriction of a fixed bound on the degree of distributional nodes. For instance, for a fixed integer $b \geq 2$, the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b}$ is the subset of $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ comprising the p-documents such that every distributional node has at most b children. We consider (o- and v-) translations between families of p-documents under this restriction (e.g., can we efficiently change a p-document to meet this restriction while preserving equivalence?). In particular, we show that for all $b_2 > b_1 \geq 2$, there is an efficient v-translation from $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_2}$ to $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_1}$ and from $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_1}$ (and $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_2}$) to $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}}$; interestingly, these are the only cases for which we show that there are efficient v-translations, but o-translations do not exist at all.

We also investigate whether families of p-documents are closed under updates. An update is naturally defined on the px-space associated with a p-document, but we would like to perform it efficiently on the p-document itself (without introducing additional types of distributional nodes). We consider tractability of updates under both the object semantics and the value semantics, but now the main difference between the two is in the language defining updates, which is richer in the case of the value semantics. We show that under the object semantics, updates are tractable in all "reasonable" models. Under the value semantics, insertions (even just those defined by single-path queries) are intractable in $\mathsf{PrXML}^{\{\mathsf{exp}\}}$, but can be done efficiently in $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ provided that they are defined by monotone queries.

After presenting some preliminaries in Section 2, we introduce p-documents in Section 3. Five types of distributional nodes are defined in Section 4, and we show that they extend the models of probabilistic XML that have been described in the literature. In Sections 5 and 6, we present results on translations between models. Updates are the subject of Section 7.

This paper extends work reported in [1, 2]. In particular, it expands the results of [1] on expressiveness of probabilistic XML models. A companion paper [10] extends the results of [1] (and some of those reported in [9]) about query evaluation over probabilistic XML models.

2 Preliminaries

We represent (probabilistic) data by unordered, unranked, labeled trees. Given a tree T, the set of nodes and the set of edges are denoted by $\mathscr{V}(T)$ and $\mathscr{E}(T)$, respectively. Note that $\mathscr{E}(T) \subseteq \mathscr{V}(T) \times \mathscr{V}(T)$. We use $\mathrm{root}(T)$ to denote the root of T. If $(n_1, n_2) \in \mathscr{E}(T)$, then n_2 is a *child* of n_1, which in turn is the *parent* of n_2. A *leaf* of T is a node without any children. Suppose that there is a path from node n_1 to node n_2. We say that n_2 is a *descendant* of n_1, whereas n_1 is an *ancestor* of n_2. Note that every node is both a descendant and an ancestor of itself. If $n_1 \neq n_2$, then n_2 is a *proper descendant* of n_1, which in turn is a *proper ancestor* of n_2. We say that the tree T' is a *subtree* of the tree T if $\mathscr{V}(T') \subseteq \mathscr{V}(T)$ and $\mathscr{E}(T') \subseteq \mathscr{E}(T)$. If T' also contains the root of T, then it is an *r-subtree* of T.

An *XML document* (a *document* for short) is a tree with a *label* attached to each node. We do not distinguish here between a tag and a value. Our notion of a label is meant to capture both. Usually, we use d to denote documents, and u, v and w to denote nodes of documents. The label of a node v is denoted by $lbl(v)$. As an example, Figure 2.1 (bottom-right) depicts a document d. Each node is represented as $[i]l$, where i is a unique identifier and l is a label. For instance, the label of Node 19 is "manager" while that of Node 14 is "*Emma*." In the figures, labels corresponding to values (rather than tags) are in italic font.

Two documents d_1 and d_2 are *isomorphic*, denoted by $d_1 \sim d_2$, if one can be obtained from the other by replacing nodes with some other nodes while preserving labels (but not identifiers). Formally, $d_1 \sim d_2$ if there is a bijection $\varphi : \mathscr{V}(d_1) \to \mathscr{V}(d_2)$, such that *(1)* for all $v \in \mathscr{V}(d_1)$ it holds that $lbl(v) = lbl(\varphi(v))$, *and (2)* for all $v_1, v_2 \in \mathscr{V}(d_1)$ we have that $(v_1, v_2) \in \mathscr{E}(d_1)$ if and only if $(\varphi(v_1), \varphi(v_2)) \in \mathscr{E}(d_2)$.

3 Probabilistic XML and p-Documents

A *probabilistic XML space* (abbr. px-space) is a probability distribution over a space of ordinary documents. Formally, it is a pair $(\mathscr{D}, p)$, where $\mathscr{D}$ is a nonempty, finite set of documents and $p : \mathscr{D} \to \mathbb{R}^+$ maps every document $d \in \mathscr{D}$ to a positive real number $p(d)$, such that $\sum_{d \in \mathscr{D}} p(d) = 1$.

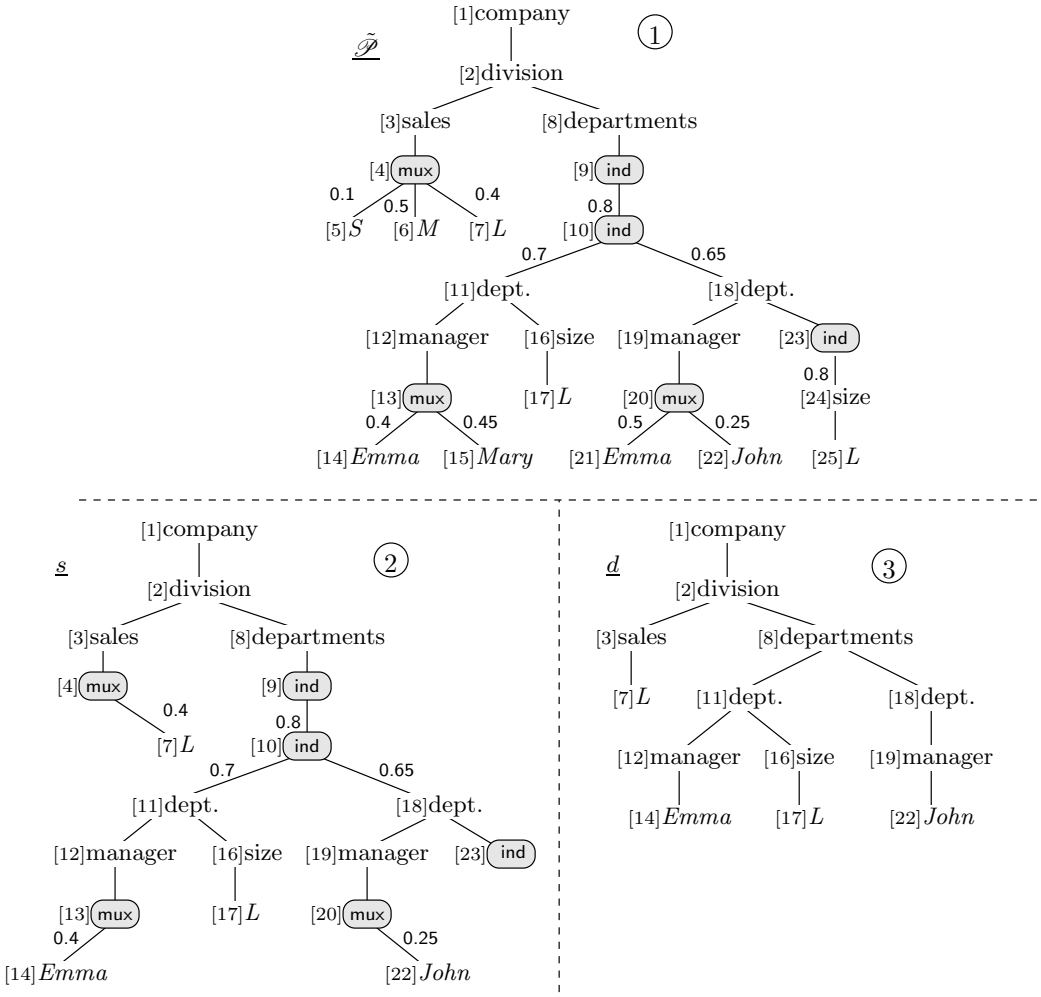

Fig. 2.1 A p-document $\tilde{\mathscr{P}}$, an r-subtree $s \in \Sigma(\tilde{\mathscr{P}})$ and the document $d = \mathsf{doc}(s)$

Typically, a px-space contains a large number of documents, so it is usually impractical to use its explicit representation (i.e., $\mathscr{D}$ and p). In this section, we show how to represent a px-space by means of a p-document, which is (a description of) a probabilistic process that generates a *random* document; that is, this process generates a document $d \in \mathscr{D}$ with probability $p(d)$.

Formally, a *p-document* is a tree $\tilde{\mathscr{P}}$ that consists of two types of nodes. *Ordinary* nodes have labels (namely, they are regular XML nodes), and they may appear in documents. *Distributional* nodes are only used for defining the probabilistic process that generates random documents (but they do not actually occur in those documents). We denote by $\mathscr{V}^{\mathrm{ord}}(\tilde{\mathscr{P}})$ and $\mathscr{V}^{\mathrm{dst}}(\tilde{\mathscr{P}})$ the disjoint sets of ordinary and distributional nodes of $\tilde{\mathscr{P}}$, respectively. The root and leaves of $\tilde{\mathscr{P}}$ are required to be ordinary nodes.

Example 3.1 Figure 2.1 (top) depicts a p-document $\tilde{\mathscr{P}}$. Distributional nodes are shown as rounded-corner rectangles. The types of those nodes are denoted by words inside the rectangles (e.g., ind and mux), and they will be discussed in Section 4.1. □

In Section 4, we define several types of distributional nodes. For now, it is sufficient to realize that each distributional node v has a probability distribution over (subsets of) its children. In the probabilistic process that generates a random document, a subset of the children of v is randomly chosen according to the distribution specified for v.

3.1 The Probabilistic Process of a p-Document

A random document of a p-document $\tilde{\mathscr{P}}$ is generated in two steps. In the first step, one subset of children is randomly chosen for each distributional node. Note that choices made for different nodes could be dependent. All the unchosen children and their descendants

(even descendants that have been chosen by their own parents) are deleted. The result is an r-subtree s of $\tilde{\mathscr{P}}$.

The second step removes all the distributional nodes. If an ordinary node u no longer has a parent, then the new parent of u is the lowest node that is both ordinary and a proper ancestor of u. The resulting document is ordinary and denoted by $\mathsf{doc}(s)$.

In terms of formal probability theory, a p-document defines the probability space that comprises the documents obtained by all the combinations of choosing for each distributional node, a subset of its children (and then removing distributional nodes, as described above). As explained in Section 4, the probability of each combination depends on the types of the distributional nodes and the probability distributions specified for those nodes (moreover, it also depends on the probabilistic dependencies that exist among those nodes).

The above pair of steps for generating a random document can be described by two *random variables* as follows. Let $\Sigma(\tilde{\mathscr{P}})$ denote the set of all the r-subtrees s of $\tilde{\mathscr{P}}$, such that every ordinary node u of s has the same set of children in both s and $\tilde{\mathscr{P}}$. The first step above chooses an r-subtree $s \in \Sigma(\tilde{\mathscr{P}})$, and we use the random variable $\mathscr{P}^{\Sigma}$ to denote that choice (i.e., s). The second step generates the document $\mathsf{doc}(\mathscr{P}^{\Sigma})$, and this document is denoted[3] by the random variable $\mathscr{P}$. Note that $\mathscr{P}$ is deterministically determined by $\mathscr{P}^{\Sigma}$.

Note that the operation $\mathsf{doc}(\cdot)$ is not necessarily one-to-one; that is, two different r-subtrees s_1 and s_2 may yield the same document. This follows from two facts: A distributional node can have a distributional child, and an empty subset of children might be selected for a distributional node.

Let $s \in \Sigma(\tilde{\mathscr{P}})$ be given. $\Pr(\mathscr{P}^{\Sigma} = s)$ is the probability that each distributional node of s chooses the exact set of children that it has in s. Thus, the probability of a random document d is given by

$$\Pr(\mathscr{P} = d) = \sum_{\substack{s \in \Sigma(\tilde{\mathscr{P}}),\\ \mathsf{doc}(s) = d}} \Pr\left(\mathscr{P}^{\Sigma} = s\right).$$

Note that $\Pr(\mathscr{P} = d)$ could be 0. In particular, the above equation implies that $\Pr(\mathscr{P} = d) = 0$ if d cannot be obtained from $\tilde{\mathscr{P}}$, that is, there is no $s \in \Sigma(\tilde{\mathscr{P}})$ such that $\Pr(\mathscr{P}^{\Sigma} = s) > 0$ and $d = \mathsf{doc}(s)$. For example, if d has a node that does not appear in $\tilde{\mathscr{P}}$, then $\Pr(\mathscr{P} = d) = 0$.

Example 3.2 Consider again Figure 2.1. Recall that the p-document $\tilde{\mathscr{P}}$ is discussed in Example 3.1. The bottom part of the figure depicts two trees. The one on the left is an r-subtree $s \in \Sigma(\tilde{\mathscr{P}})$, and the one on the right is the document $d = \mathsf{doc}(s)$. It can be easily shown that s is the only r-subtree of $\Sigma(\tilde{\mathscr{P}})$ that generates d and, consequently, $\Pr(\mathscr{P} = d) = \Pr(\mathscr{P}^{\Sigma} = s)$. The computation of the probability on the right-hand side will be explained in Section 4.1. □

The *possible worlds* of a p-document $\tilde{\mathscr{P}}$ are all the documents with a nonzero probability, i.e., documents d, such that $\Pr(\mathscr{P} = d) > 0$. We use $pwd(\tilde{\mathscr{P}})$ to denote the set of all the possible worlds. Clearly,

$$\sum_{d \in pwd(d)} \Pr(\mathscr{P} = d) = 1.$$

To conclude, a p-document $\tilde{\mathscr{P}}$ defines the px-space $(\mathscr{D}, p)$, where $\mathscr{D}$ is the set $pwd(\tilde{\mathscr{P}})$ and p is the function $\Pr(\mathscr{P} = \cdot)$. We use $[\![\tilde{\mathscr{P}}]\!]$ to denote this px-space.

3.2 Isomorphism and Equivalence

Two px-spaces $(\mathscr{D}_1, p_1)$ and $(\mathscr{D}_2, p_2)$ are *isomorphic,* denoted by $(\mathscr{D}_1, p_1) \sim (\mathscr{D}_2, p_2)$, if they are identical *up to isomorphism*, that is, for all documents d,

$$\sum_{d' \in \mathscr{D}_1 | d' \sim d} p_1(d') = \sum_{d' \in \mathscr{D}_2 | d' \sim d} p_2(d').$$

In words, for all documents d, the probability that a document of $(\mathscr{D}_1, p_1)$ is isomorphic to d is equal to the probability that a document of $(\mathscr{D}_2, p_2)$ is isomorphic to d.

Two p-documents are *equivalent* if they define the same px-space. There are two variants of equivalence depending on whether two px-spaces are deemed the same based on equality or isomorphism. The first notion of equivalence follows the *object-based* semantics, whereas the second uses the *value-based* semantics. The formal definitions follow.

Two p-documents $\tilde{\mathscr{P}}_1$ and $\tilde{\mathscr{P}}_2$ are *o-equivalent*, denoted by $\tilde{\mathscr{P}}_1 \equiv_o \tilde{\mathscr{P}}_2$, if $[\![\tilde{\mathscr{P}}_1]\!] = [\![\tilde{\mathscr{P}}_2]\!]$; namely, for all documents d, we have that $\Pr(\mathscr{P}_1 = d) = \Pr(\mathscr{P}_2 = d)$. Analogously, $\tilde{\mathscr{P}}_1$ and $\tilde{\mathscr{P}}_2$ are *v-equivalent*, denoted by $\tilde{\mathscr{P}}_1 \equiv_v \tilde{\mathscr{P}}_2$, if $[\![\tilde{\mathscr{P}}_1]\!] \sim [\![\tilde{\mathscr{P}}_2]\!]$, namely, $\Pr(\mathscr{P}_1 \sim d) = \Pr(\mathscr{P}_2 \sim d)$ holds for all documents d. Observe that if $\tilde{\mathscr{P}}_1 \equiv_o \tilde{\mathscr{P}}_2$, then $\tilde{\mathscr{P}}_1$ and $\tilde{\mathscr{P}}_2$ have the same set of possible worlds; however, this does not necessarily hold if $\tilde{\mathscr{P}}_1 \equiv_v \tilde{\mathscr{P}}_2$. Clearly, object equivalence implies value equivalence,[4] but not vice versa.

Observe that if $\tilde{\mathscr{P}}_1 \equiv_o \tilde{\mathscr{P}}_2$, then their sets of ordinary nodes are identical. More precisely, either $\tilde{\mathscr{P}}_1$ or $\tilde{\mathscr{P}}_2$ may have an ordinary node that does not appear in

[3] Note that $\tilde{\mathscr{P}}$ denotes a p-document, whereas $\mathscr{P}$ (i.e., without the tilde) is the random variable that denotes a document generated from $\tilde{\mathscr{P}}$ by the two-step probabilistic process.

[4] By definition, if a node of a p-document exists in two different documents, then it has the same label in both.

the other one if the probability of choosing that node is zero; however, such nodes are useless and can always be eliminated.

4 Families of Concrete p-Documents

In this section, we define several types of distributional nodes. A concrete p-document is obtained by specifying the types and probability distributions of the distributional nodes. Later in this section, we discuss families of p-documents. A specific family is characterized by two properties: *(1)* the types of distributional nodes that are allowed in the p-documents, and *(2)* whether one can construct hierarchies consisting of only distributional nodes. We explain how our framework gives rise to a variety of models of probabilistic XML, including most (if not all) of the models that have been studied in the literature.

4.1 Types of Distributional Nodes

To obtain a concrete p-document, we should specify for each distributional node v, the probability distribution of choosing a subset of the children of v. We define five types of distributional nodes, each with a different way of describing that probability distribution.

Type ind (for *independent*)**.** A node v of type ind specifies for every child w, the probability $p^v(w)$ of choosing w; this choice is independent of any other choice of children (of either v or other distributional nodes). Hence, the probability of choosing a subset C of children of v is

$$\prod_{w \in C} p^v(w) \prod_{w \in \bar{C}} (1 - p^v(w)),$$

where $\bar{C}$ is the set of children of v that are not in C.

Type mux (for *mutually exclusive*)**.** A node v of type mux specifies the probabilities $p^v(w_1), \ldots, p^v(w_k)$ for its children $w_1, \ldots, w_k$, respectively. Node v chooses at most one child w_i with the probability $p^v(w_i)$, independently of the other distributional nodes. We require that $\sum_{i=1}^{k} p^v(w_i) \leq 1$. The probability that v chooses none of its children is $1 - \sum_{i=1}^{k} p^v(w_i)$.

Type det (for *deterministic*)**.** A node v of type det always chooses all of its children, namely, each child is chosen with probability 1.

Type exp (for *explicit*)**.** A node v of type exp specifies probabilities $p^v(W_1), \ldots, p^v(W_l)$, where the W_i are some (but not necessarily all of the) distinct subsets of the children of v. Node v chooses exactly one subset W_i with the probability $p^v(W_i)$, independently of the other distributional nodes. Note that one of the W_i may be empty. We require that $\sum_{i=1}^{l} p^v(W_i) = 1$.

Type cie (for *conjunction of independent events*)**.** In a given p-document, nodes of this type are associated with independent random Boolean variables $e_1, \ldots, e_m$, called *events*. For each event e_i, the p-document specifies the probability $p(e_i)$ that e_i is **true**. A node v of type cie specifies for every child w, a conjunction $\alpha^v(w) = a_1 \wedge \cdots \wedge a_{k_w}$ ($k_w > 0$), where each a_j is either e_i or $\neg e_i$ for some $1 \leq i \leq m$. Note that different conjunctions can share common events, and the number of events in $\alpha^v(w)$ (i.e., k_w) may vary from one child of v to another. Before generating a document, values for $e_1, \ldots, e_m$ are randomly determined. A child w is chosen if its corresponding conjunction $\alpha^v(w)$ is **true**.

Note that if the type of a distributional node v is one of the first four (i.e., ind, mux, det or exp), then v randomly picks children independently of the probabilistic choices made by the other distributional nodes of the p-document. But different distributional nodes of type cie can *correlate* their choices by sharing events.

Example 4.1 The p-document $\tilde{\mathscr{P}}$ of Figure 2.1 has ind and mux nodes. The probability specified for each child is shown next to the edge that leads to that child. We now describe how to compute the probability $\Pr(S = s)$ of the document $s \in \Sigma(\tilde{\mathscr{P}})$ that is shown in the bottom-left part of Figure 2.1. Each mux node of s chooses exactly one child with the probability specified for that child. The probabilities of the choices made by the ind nodes are as follows. Node 9 chooses its only child with probability 0.8. Node 10 chooses both children with probability $0.7 \cdot 0.65 = 0.455$. And Node 23 chooses the empty set of children with probability $1 - 0.8 = 0.2$. $\Pr(S = s)$ is the product of the probabilities of the choices made by all the distributional nodes. □

A node v of a p-document is *useless* if there is no r-subtree s of $\tilde{\mathscr{P}}$, such that $\Pr\left(\mathscr{P}^{\Sigma} = s\right) > 0$ and v appears in s. One can efficiently find all the useless nodes of a p-document and delete them (as well as their descendants). If, as a result, a distributional node has no ordinary descendants, then it is also removed. In practice, it is not necessary to remove useless nodes; however, we assume that p-documents do not have useless nodes, because it is needed in some of the proofs.

We denote by $\mathsf{PrXML}^{\{\text{type}_1,\text{type}_2,\ldots\}}$ the family of all the p-documents, such that the types of their distributional nodes are among those listed in the superscript. For example, the p-documents of $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ use only ind and mux nodes.

In the following, our complexity analysis makes an implicit assumption that numbers (e.g., probabilities of

the form $p^v(w_i)$ or $p(e_i)$ specified in p-documents) are represented in a way that the basic arithmetic operations (e.g., computing the product or sum of a series of numbers) can be performed efficiently.

4.2 Hierarchy of Distributional Nodes

The straightforward way of using a distributional node is when both its parent and children are ordinary nodes. In this case, the role of the distributional node is to choose ordinary children for its ordinary parent. Sometimes, however, we can obtain more complex distributions (over the probability space of documents) by constructing hierarchies of distributional nodes.

If every distributional node of a p-document $\tilde{\mathscr{P}}$ has only ordinary children, we say that $\tilde{\mathscr{P}}$ is *distributional-hierarchy free* (abbr. DHF). As an example, consider Figure 2.1. The p-document $\tilde{\mathscr{P}}$ is not DHF, because Node 10 is the child of Node 9 and both are distributional nodes. If $\mathscr{F}$ is a set of p-documents, then $\mathscr{F}_{|\not\mathcal{H}}$ denotes the restriction of $\mathscr{F}$ to its DHF p-documents.

In Section 5, we show that in some families of p-documents, we can express more px-spaces by allowing hierarchies of distributional nodes.

4.3 Previously Studied Models

The family $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ is the same as the ProTDB documents of [3]. The probabilistic XML model[5] of [6] is a subset of $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$, where mux nodes (called "probability nodes") have as children only det nodes (called "possibility nodes") and det nodes have only ordinary children (called "XML nodes").

The model of probabilistic XML that was investigated in [2, 7] is $\mathsf{PrXML}^{\{\mathsf{cie}\}}_{|\not\mathcal{H}}$. In the next section, we show that adding hierarchies of distributional nodes to $\mathsf{PrXML}^{\{\mathsf{cie}\}}_{|\not\mathcal{H}}$ is not needed (that is, every p-document of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ can be efficiently translated to a p-document of $\mathsf{PrXML}^{\{\mathsf{cie}\}}_{|\not\mathcal{H}}$). The "simple probabilistic trees" of [7] are actually the family $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\mathcal{H}}$ (hierarchies make a difference in this case). The same is true for the probabilistic XML model underlying the "PEPX" system [11].

The work of [4] introduced a model of probabilistic XML graphs, where each node explicitly specifies the probability distribution over its possible sets of children. Restricting their XML graphs to trees yields a sub-family of $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\mathcal{H}}$ (a lack of hierarchies is significant when only exp nodes are allowed). The same is true for [5] if we restrict their intervals to points.

With respect to probabilistic relational models [12–14], the comparison is more delicate because there has been a lot of research in this direction, some of it not relevant here. (In particular, a large part deals with query processing or the origins of imprecision.) From a modeling viewpoint, one can easily represent a relation as an XML tree with a node for each tuple and a node for each entry in a tuple. Distributional nodes can then be used to specify probabilities on tuples and on values inside tuples. For the relational model, the notion of probabilistic possible worlds has also been used and many representation systems have been proposed. The block-independent model of [12] (which is an incomplete representation system) can be translated into the family $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ in a straightforward way. Other probabilistic relational models (in particular, [13]) can be seen as probabilistic versions of the conditional tables of [15]. (In that direction, one most elaborate work is that of [16].) In some sense, the $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ model generalizes this idea to trees, and the main features of corresponding probabilistic relational models can accordingly be represented in this probabilistic XML model. For instance, the lineage of Trio [13] can naturally be encoded as independent events.[6] A general study of the translation of existing probabilistic relational models into probabilistic XML models is an interesting issue, but beyond the scope of this paper.

5 Translations Between Families of P-Documents

The previous section described several families of p-documents. In this section, we compare the expressive power of these families. We first formalize the notion of expressive power.

5.1 Translators

Consider two (infinite) sets $\mathscr{F}_1$ and $\mathscr{F}_2$ of p-documents. We say that $\mathscr{F}_1$ is *o-translatable* to $\mathscr{F}_2$, denoted by $\mathscr{F}_1 \sqsubseteq_o \mathscr{F}_2$, if each document of $\mathscr{F}_1$ is o-equivalent to some document of $\mathscr{F}_2$. That is, for each document $\tilde{\mathscr{P}}_1 \in \mathscr{F}_1$, there exists a document $\tilde{\mathscr{P}}_2 \in \mathscr{F}_2$, such that $\tilde{\mathscr{P}}_1 \equiv_o \tilde{\mathscr{P}}_2$. An *o-translator* from $\mathscr{F}_1$ to $\mathscr{F}_2$ is

[5] In the probabilistic documents of [6], the root is distributional. We can assume that a dummy ordinary node is added for compliance with the definition of p-documents.

[6] Note that Trio allows annotating tuples with arbitrary propositional formulas. An efficient translation into a $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ tree requires such formulas to be in DNF. Allowing arbitrary formulas as conditions on distributional nodes makes query processing less efficient, as discussed in [8].

an algorithm that receives as input a $\tilde{\mathscr{P}}_1 \in \mathscr{F}_1$ and generates an o-equivalent $\tilde{\mathscr{P}}_2 \in \mathscr{F}_2$. If there is an efficient o-translator from $\mathscr{F}_1$ to $\mathscr{F}_2$ (i.e., a translator that runs in polynomial time in the size of its input $\tilde{\mathscr{P}}_1$), then $\mathscr{F}_1$ is *efficiently o-translatable* to $\mathscr{F}_2$, denoted by $\mathscr{F}_1 \sqsubseteq_o^{\text{poly}} \mathscr{F}_2$. If $\mathscr{F}_1 \sqsubseteq_o \mathscr{F}_2$ and $\mathscr{F}_2 \sqsubseteq_o \mathscr{F}_1$, then we write $\mathscr{F}_1 \equiv_o \mathscr{F}_2$. Similarly, $\mathscr{F}_1 \equiv_o^{\text{poly}} \mathscr{F}_2$ means that there are efficient o-translators in both directions.

We use analogous definitions and notation for the notion of *v-translation*. As an example, $\mathscr{F}_1 \sqsubseteq_v^{\text{poly}} \mathscr{F}_2$ means that there is an efficient *v-translator* that receives as input a $\tilde{\mathscr{P}}_1 \in \mathscr{F}_1$ and generates a $\tilde{\mathscr{P}}_2 \in \mathscr{F}_2$, such that $\tilde{\mathscr{P}}_1 \equiv_v \tilde{\mathscr{P}}_2$.

5.2 The Types ind, mux and det

In this section, we consider the three types ind, mux and det. We first study the families that use only one of these three types.

5.2.1 Using Each Type Individually

Using only distributional nodes of type det is, obviously, meaningless in the sense that the resulting p-document is deterministic. Formally, the px-space defined by a p-document $\tilde{\mathscr{P}}$ of the family $\mathsf{PrXML}^{\{\mathsf{det}\}}$ consists of only one document, namely, $\mathsf{doc}(\tilde{\mathscr{P}})$. Consequently, $\mathsf{PrXML}^{\{\mathsf{det}\}} \equiv_o^{\text{poly}} \mathsf{PrXML}^{\{\}}$, which means that the family $\mathsf{PrXML}^{\{\mathsf{det}\}}$ is trivially o-translatable to any other family (among those we consider).

Next, we show that hierarchy is not required in the family $\mathsf{PrXML}^{\{\mathsf{mux}\}}$.

Lemma 5.1 $\mathsf{PrXML}^{\{\mathsf{mux}\}} \equiv_o^{\text{poly}} \mathsf{PrXML}^{\{\mathsf{mux}\}}_{|\not\downarrow}$.

Proof Let $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{mux}\}}$ be given. We efficiently transform $\tilde{\mathscr{P}}$ into an o-equivalent p-document $\tilde{\mathscr{P}}' \in \mathsf{PrXML}^{\{\mathsf{mux}\}}_{|\not\downarrow}$ by repeatedly eliminating each distributional node u that has a distributional parent, until there is no such node (and, thus, the p-document is DHF). The elimination process is the following. Consider two distributional nodes u and v of $\tilde{\mathscr{P}}$, such that v is the parent of u (and, of course, both u and v are of type mux). Let $w_1, \ldots, w_k$ be the children of u. We remove u from $\tilde{\mathscr{P}}$ and connect every w_i to v (i.e., w_i becomes a child of v). For all $1 \leq i \leq k$, the probability $p^v(w_i)$ is set to $p^v(u) \cdot p^u(w_i)$. Observe that each step of the elimination process preserves o-equivalence; hence, this transformation is correct. □

Unlike $\mathsf{PrXML}^{\{\mathsf{mux}\}}$, hierarchy is essential in the family $\mathsf{PrXML}^{\{\mathsf{ind}\}}$. In particular, the following lemma shows that $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ is not v-translatable (and, hence, not o-translatable) to $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$ (regardless of efficiency).

Lemma 5.2 $\mathsf{PrXML}^{\{\mathsf{ind}\}} \not\sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$.

Proof Let $\tilde{\mathscr{P}}$ denote the p-document of $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ that is depicted in Figure 5.1(a). Note that the ordinary nodes w_1 and w_2 of $\tilde{\mathscr{P}}$ are labeled with a and b, respectively. We will prove that there is no DHF p-document of $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$ that is v-equivalent to $\tilde{\mathscr{P}}$. Suppose, by way of contradiction, that $\tilde{\mathscr{P}}'$ is such a p-document. Observe that none of the probabilities specified in $\tilde{\mathscr{P}}'$ is zero, because there are no useless nodes. Therefore, $\tilde{\mathscr{P}}'$ has a possible world that includes all of its ordinary nodes. Consequently, the assumption $\tilde{\mathscr{P}}' \equiv_v \tilde{\mathscr{P}}$ implies that $\tilde{\mathscr{P}}'$ must have the following three properties. First, the root of $\tilde{\mathscr{P}}'$ has exactly two ordinary descendants. Second, one of these two nodes, denoted by u_a, is labeled with a and the other, denoted by u_b, is labeled with b. Third, neither one of u_a and u_b is an ancestor of the other (because some possible world of $\tilde{\mathscr{P}}$ contains both w_1 and w_2 as siblings). Note that each of u_a and u_b is either a child or a grandchild of the root, because $\tilde{\mathscr{P}}'$ is DHF. It follows that the probabilistic events "$\mathscr{P}'$ includes the label a" and "$\mathscr{P}'$ includes the label b" are independent. However, this is not the case for $\mathscr{P}$, because the probability that $\mathscr{P}$ includes both a and b is 0.5^3 whereas the probabilities of the events "$\mathscr{P}$ includes a" and "$\mathscr{P}$ includes b" are both 0.5^2. This contradicts the v-equivalence of $\tilde{\mathscr{P}}$ and $\tilde{\mathscr{P}}'$. □

Next, we consider the relationships between families that use different types of distributional nodes. The first lemma below gives a negative result, namely, p-documents with only mux nodes and no hierarchies are not v-translatable to p-documents that use only ind nodes. The second lemma states a positive result for the opposite direction; that is, p-documents with only ind nodes and no hierarchies are efficiently o-translatable (and v-translatable) to p-documents that use only mux nodes.

Lemma 5.3 $\mathsf{PrXML}^{\{\mathsf{mux}\}}_{|\not\downarrow} \not\sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{ind}\}}$.

Proof The lemma holds because $\mathsf{PrXML}^{\{\mathsf{mux}\}}_{|\not\downarrow}$, but not $\mathsf{PrXML}^{\{\mathsf{ind}\}}$, contains a p-document that has two labels l_1 and l_2, such that *(1)* each of l_1 and l_2 appears in one or more possible worlds, *and (2)* no possible world contains both l_1 and l_2. □

Lemma 5.4 $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow} \sqsubseteq_o^{\text{poly}} \mathsf{PrXML}^{\{\mathsf{mux}\}}$.

Proof To prove the lemma, we describe an efficient o-translator from $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$ to $\mathsf{PrXML}^{\{\mathsf{mux}\}}_{|\not\downarrow}$. Consider a p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$ and let v be an ind node of $\tilde{\mathscr{P}}$. Suppose that the parent of v is u and the children of v are $w_1, \ldots, w_k$. The nodes u and $w_1, \ldots, w_k$ are

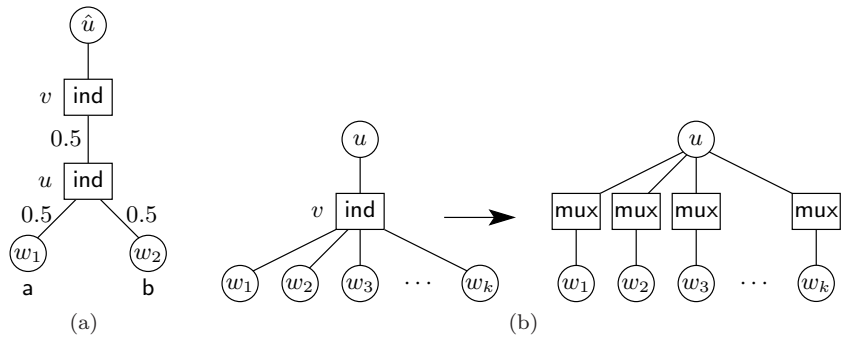

Fig. 5.1 (a) A p-document of $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ that cannot be v-translated to $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{\not|\not|}$ (b) O-translating $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{\not|\not|}$ to $\mathsf{PrXML}^{\{\mathsf{mux}\}}$

ordinary, because $\tilde{\mathscr{P}}$ is DHF. We replace v with k new mux nodes $v_1, \ldots, v_k$, as illustrated in Figure 5.1(b). Each v_i is a child of u and the parent of w_i. For $1 \leq i \leq k$, we define $p^{v_i}(w_i) = p^v(w_i)$. □

The following lemma shows that the previous result no longer holds if we allow hierarchies of ind nodes; furthermore, $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ is not even v-translatable to $\mathsf{PrXML}^{\{\mathsf{mux}\}}$.

Lemma 5.5 $\mathsf{PrXML}^{\{\mathsf{ind}\}} \not\sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{mux}\}}$.

Proof Recall the proof of Lemma 5.2 and, in particular, consider again the p-document $\tilde{\mathscr{P}}$ (which is depicted in Figure 5.1(a)). To derive a contradiction, we use Lemma 5.1 and assume that $\mathsf{PrXML}^{\{\mathsf{mux}\}}_{\not|\not|}$ has a p-document $\tilde{\mathscr{P}}'$ that is v-equivalent to $\tilde{\mathscr{P}}$. All the probabilities specified in $\tilde{\mathscr{P}}'$ are nonzero, because there are no useless nodes.

The root r of $\tilde{\mathscr{P}}'$ cannot have ordinary children, because there is a possible world of $\tilde{\mathscr{P}}$ that consists of a single node. Hence, all the children of r are mux nodes and each of them has only ordinary children (because $\tilde{\mathscr{P}}'$ is DHF).

The same label (i.e., either a or b) cannot appear under two distinct mux children of r, or else some possible world of $\tilde{\mathscr{P}}'$ contains more than one occurrence of that label. Let u_a and u_b be the mux nodes of $\tilde{\mathscr{P}}'$ that have, among their children, all the nodes with the labels a and b, respectively. If u_a and u_b are distinct, then the probabilistic events "$\mathscr{P}'$ includes the label a" and "$\mathscr{P}'$ includes the label b" are independent. Hence, as in the proof of Lemma 5.2, this contradicts the assumption that $\tilde{\mathscr{P}} \equiv_v \tilde{\mathscr{P}}'$. If $u_a = u_b$, then no possible world of $\tilde{\mathscr{P}}'$ contains both a and b which, again, contradicts $\tilde{\mathscr{P}} \equiv_v \tilde{\mathscr{P}}'$. □

The following theorem summarizes this section.

Theorem 5.6 *The following hold.*

1. $\mathsf{PrXML}^{\{\mathsf{mux}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{mux}\}}_{\not|\not|}$.
2. $\mathsf{PrXML}^{\{\mathsf{ind}\}} \not\sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{ind}\}}_{\not|\not|}$.
3. $\mathsf{PrXML}^{\{\mathsf{det}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{det}\}}_{\not|\not|} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\}}$.
4. $\mathsf{PrXML}^{\{\mathsf{mux}\}} \not\sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{ind}\}}$ *and* $\mathsf{PrXML}^{\{\mathsf{mux}\}} \not\sqsupseteq_v \mathsf{PrXML}^{\{\mathsf{ind}\}}$.
5. $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{\not|\not|} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{mux}\}}$.

5.2.2 Combinations of the Types ind, mux *and* det

We now consider the families that use at least two of the types ind, mux and det. Observe that the type det is a special case of ind (i.e., each child is chosen with probability 1). Therefore, adding the type det does not change the expressive power of a family that is allowed to use ind nodes. In particular, $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{det}\}}$. (Recall that $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ is the same as the ProTDB model [3].)

We first show that under the value-based semantics, the family $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$ can represent every px-space consisting of documents that have the same label in their roots. Formally, a px-space $(\mathscr{D}, p)$ is *root consistent* if for every two documents $d_1, d_2 \in \mathscr{D}$, it holds that $lbl(\mathrm{root}(d_1)) = lbl(\mathrm{root}(d_2))$.

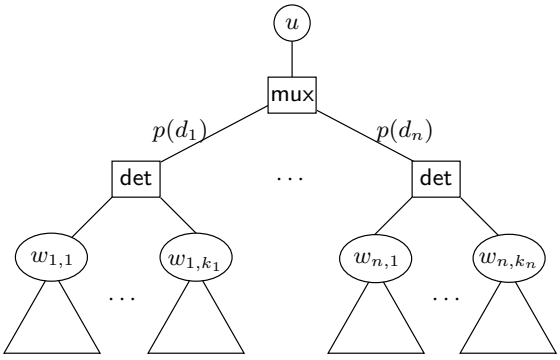

Fig. 5.2 Transforming a px-space into a hierarchy of det and mux nodes

Proposition 5.7 *For all the root-consistent px-spaces $(\mathscr{D}, p)$, there exists a p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$ such that $(\mathscr{D}, p) \sim [\![\tilde{\mathscr{P}}]\!]$. As a special case, for every subset $C \subseteq \{\mathsf{ind}, \mathsf{mux}, \mathsf{det}, \mathsf{exp}, \mathsf{cie}\}$, it holds that $\mathsf{PrXML}^{C} \sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$.*

Proof Let $(\mathscr{D}, p)$ be a px-space, where $\mathscr{D} = \{d_1, \ldots, d_n\}$. For each i, let $w_{i,1}, \ldots, w_{i,k_i}$ be the k_i children of the root of d_i. We construct the p-document $\tilde{\mathscr{P}}$ shown in Figure 5.2. Namely, u is a new node that has the same label as the roots of $d_1, \ldots, d_n$, and its only child is a new mux node. Each d_i becomes a subtree of the mux node after replacing its root with a new det node, which is chosen by the mux node with probability $p(d_i)$. Clearly, $(\mathscr{D}, p)$ is isomorphic to $[\![\tilde{\mathscr{P}}]\!]$. □

Note that the above proof constructs a p-document having a size that is linear in the given px-space $(\mathscr{D}, p)$.

The following lemma shows that just by adding det nodes to the family $\mathsf{PrXML}^{\{\mathsf{mux}\}}$, we get the expressive power of all the three types det, ind, and mux. This lemma and Part 4 of Theorem 5.6 imply that $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}} \not\sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{mux}\}}$.

Lemma 5.8 $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$.

Proof $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ is trivial, because a det node is a special case of an ind node (i.e., every child is chosen with probability 1). For the other direction, $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$, let $\tilde{\mathscr{P}}$ be a p-document of $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$. We efficiently transform $\tilde{\mathscr{P}}$ into a p-document of $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$ while preserving o-equivalence as described next. Consider an ind node v of $\tilde{\mathscr{P}}$, and let $w_1, \ldots, w_k$ be the children of v. We replace v and its children with the subtree shown in Figure 5.3. That is, v is replaced with a new det node v' that has k new mux nodes $u_1, \ldots, u_k$ as children. For each u_i, the node w_i is the only child of u_i and it is chosen with probability $p^v(w_i)$. □

Now, we consider the need for hierarchies of distributional nodes when combining two or more of the three types ind, mux and det. Note that $\mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind},\mathsf{det}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind}\}}$, so this case has been studied in the previous section. Also observe that $\mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind},\mathsf{mux},\mathsf{det}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind},\mathsf{mux}\}}$. The following lemma shows that in p-documents without hierarchies of distributional nodes, ind is not needed if mux is used.

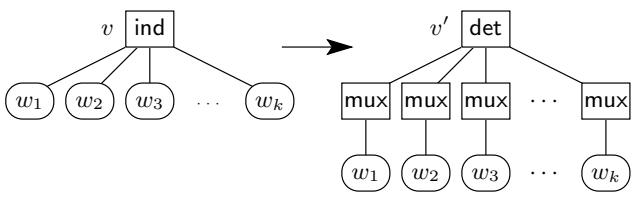

Fig. 5.3 Transforming an ind node into a hierarchy of det and mux nodes

Lemma 5.9 $\mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind},\mathsf{mux}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}_{\not|\not/}^{\{\mathsf{mux}\}}$.

Proof The proof of Lemma 5.4 shows that if both the parent and the children of an ind node v are ordinary, then v can be emulated by some mux nodes without introducing hierarchies. Therefore, $\mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind},\mathsf{mux}\}} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}_{\not|\not/}^{\{\mathsf{mux}\}}$. The opposite direction, $\mathsf{PrXML}_{\not|\not/}^{\{\mathsf{mux}\}} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind},\mathsf{mux}\}}$, is trivially true. □

The following theorem is an immediate corollary of Lemmas 5.5, 5.8 and 5.9.

Theorem 5.10 *The following hold.*

1. $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{det}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}} \not\sqsubseteq_v \mathsf{PrXML}^{\{\mathsf{mux}\}}$.
2. $\mathsf{PrXML}_{\not|\not/}^{\{\mathsf{ind},\mathsf{mux},\mathsf{det}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}_{\not|\not/}^{\{\mathsf{mux}\}}$.

In the remainder of this section, we omit the specification of the type det in a family that uses the type ind, because the first can be thought of as a special case of the second.

5.3 The Type exp

We now consider the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}$. Observe that the type mux is a special case of exp; that is, a mux node chooses with nonzero probability only singletons and possibly the empty set. Similarly, a node of type det is an exp node that chooses the set of all of its children with probability 1. In the proof of Lemma 5.8, we showed how an ind node is emulated by mux and det nodes. Thus, we get the following result, which implies that $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ generalizes $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$. As shown later in Lemma 5.14, this generalization is strict, namely, there is no o-translation from $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ (or even $\mathsf{PrXML}_{\not|\not/}^{\{\mathsf{exp}\}}$) to the family $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$.

Lemma 5.11 $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{exp}\}}$.

Next, we consider the need for hierarchies of distributional nodes in $\mathsf{PrXML}^{\{\mathsf{exp}\}}$. The following theorem shows that one can always eliminate hierarchies from a p-document of $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ (while preserving o- and v-equivalence), but it may cause an exponential blowup even if all the exp nodes actually emulate ind nodes. A particular consequence is that the models of [4, 5], restricted to trees with point probabilities, are not as general as $\mathsf{PrXML}^{\{\mathsf{exp}\}}$. This theorem also shows that if a p-document of $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}}$ is DHF, then it can be efficiently o-translated to a p-document of $\mathsf{PrXML}^{\{\mathsf{exp}\}}$

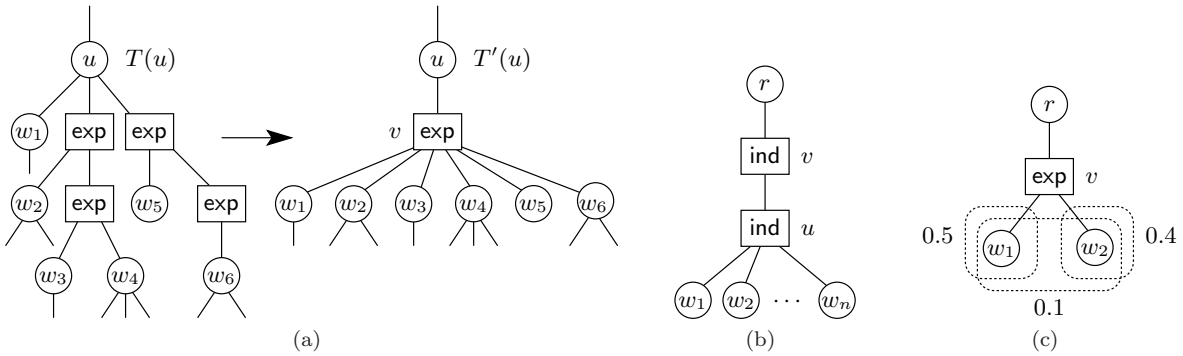

Fig. 5.4 (a) Transforming a hierarchy of exp nodes into a single exp node (b) A p-document of $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ that cannot be v-translated to $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\downarrow\not/}$ without an exponential blowup (c) A p-document that cannot be o-translated to $\mathsf{PrXML}^{\{\mathsf{cie}\}}$

without introducing hierarchies of distributional nodes (note that the construction in the proof of Lemma 5.11 does not have this property).

Theorem 5.12 *The following hold.*

1. $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}} \equiv_o \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\downarrow\not/}$.
2. $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}}_{\not\downarrow\not/} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\downarrow\not/}$.
3. $\mathsf{PrXML}^{\{\mathsf{ind}\}} \not\sqsubseteq_v^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\downarrow\not/}$.
4. $\mathsf{PrXML}^{\{\mathsf{exp}\}} \not\sqsubseteq_v^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\downarrow\not/}$.

Proof We first prove Part 1. By Lemma 5.11, it is sufficient to show that $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ is o-translatable to the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\downarrow\not/}$, and that can be done by repeatedly applying the following transformation to a $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}$. Consider an ordinary node u of $\tilde{\mathscr{P}}$ that has some exp children as well as exp grandchildren. Let $T(u)$ be the maximal subtree of $\tilde{\mathscr{P}}$, such that the root is u, all the interior nodes are distributional and all the leaves are ordinary. We replace $T(u)$ with $T'(u)$, as shown in Figure 5.4(a). That is, we remove all the interior nodes of $T(u)$, add a new exp node v as the only child of u, and each leaf of $T(u)$ becomes a child of v. For each subset W of the children of v, we define

$$p^v(W) = \Pr(W \text{ is the set of children of } u \text{ in } \mathscr{P} \mid u \in \mathscr{V}(\mathscr{P})).$$

Observe that the above translation is, in general, inefficient, since exponentially many probabilities are computed (i.e., for each of the subsets of the children of u).

For Part 2, the proof of Lemma 5.4 shows how, in a DHF p-document, we can transform an ind node to several mux nodes without creating a hierarchy, and a mux node is a special case of an exp node.

To prove Part 3, consider the p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{ind}\}}$ of Figure 5.4(b). Nodes v and u of $\tilde{\mathscr{P}}$ choose each of their children with probability 1/2. The ordinary nodes $w_1, \ldots, w_n$ have n distinct labels $l_1, \ldots, l_n$, respectively.

Suppose that some $\tilde{\mathscr{P}}' \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\downarrow\not/}$ satisfies $\tilde{\mathscr{P}}' \equiv_v \tilde{\mathscr{P}}$. The children of root($\tilde{\mathscr{P}}'$) are exp nodes and the grandchildren are ordinary nodes, because some possible world of $\tilde{\mathscr{P}}$ comprises just the root r. Each child of an exp node belongs to some subset with nonzero probability, because there are no useless nodes.

If children of distinct exp nodes have the same label l_i, then there is a document $d \in pwd(\tilde{\mathscr{P}}')$ that has two occurrences of l_i, which cannot happen in any document of $pwd(\tilde{\mathscr{P}})$, in contradiction to $\tilde{\mathscr{P}}' \equiv_v \tilde{\mathscr{P}}$. Therefore, each label occurs under exactly one exp child of root($\tilde{\mathscr{P}}'$).

Now, suppose that the labels l_i and l_j $(i \neq j)$ occur below two distinct exp nodes of $\tilde{\mathscr{P}}'$. Hence, the probabilistic events "$\mathscr{P}'$ includes the label l_i" and "$\mathscr{P}'$ includes the label l_j" are independent. However, this is not the case in documents of $pwd(\tilde{\mathscr{P}})$, because if l_i appears in a document $d \in pwd(\tilde{\mathscr{P}})$, it means that node u of $\tilde{\mathscr{P}}$ has been chosen, and therefore, the probability that l_j also appears in d is 1/2 and not 1/4. Consequently, $\tilde{\mathscr{P}}'$ has only one exp node.

Since every subset of the labels occurs in some possible world of $\tilde{\mathscr{P}}$, it follows that 2^n probabilities are specified by the exp node of $\tilde{\mathscr{P}}'$. Therefore, the size of this specification is exponential in the size of $\tilde{\mathscr{P}}$.

Part 4 follows from Part 3 and Lemma 5.11. □

5.4 The Type cie

We now discuss the expressive power of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. The following theorem proves that this family generalizes $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$; a later result in this section shows that the generalization is strict. Moreover, hierarchies of distributional nodes are not needed in $\mathsf{PrXML}^{\{\mathsf{cie}\}}$.

Theorem 5.13 $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{cie}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{cie}\}}_{|\not\perp}$.

Proof We transform a p-document $\tilde{\mathscr{P}}$ of the family $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{cie}\}}$ to a document of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ as follows. We consider every node v of $\tilde{\mathscr{P}}$, such that the type of v is either ind or mux; let $w_1, \ldots, w_k$ be the children of v. First, we change the type of v to cie and introduce k new events $e_1, \ldots, e_k$. If v is an ind node, then for all $1 \leq i \leq k$, we define $p(e_i) = p^v(w_i)$ and $\alpha^v(w_i) = e_i$. If v is a mux node, then no w_i satisfies $p^v(w_i) = 0$, because there are no useless nodes, and so we do the following. For all $1 \leq i \leq k$, we define $\alpha^v(w_i) = e_i \wedge \neg e_{i-1} \wedge \cdots \wedge \neg e_1$ and specify the probabilities $p(e_1) = p^v(w_1)$, $p(e_2) = p^v(w_2)/(1 - p^v(w_1))$ and, in general, $p(e_i) = p^v(w_i) \cdot \prod_{j=1}^{i-1}(1 - p(e_j))^{-1}$. Hence, the probability that $\alpha^v(w_i)$ is **true** is $p^v(w_i)$. To show that the probabilities are well defined, we prove that $p(e_i) < 1$ for $1 \leq i < k$. (We also have to show that $p(e_k) \leq 1$ and this is proved similarly.) Suppose otherwise and consider the smallest l, such that $p(e_l) \geq 1$ or equivalently $p^v(w_l) \geq \prod_{j=1}^{l-1}(1 - p(e_j))$. $1 - \prod_{j=1}^{l-1}(1 - p(e_j))$ is the probability that at least one of the events $e_1, \ldots, e_{l-1}$ is **true** or, equivalently, exactly one of $\alpha^v(w_1), \ldots, \alpha^v(w_{l-1})$ is **true**. The $\alpha^v(w_i)$ are disjoint events and therefore $1 - \prod_{j=1}^{l-1}(1 - p(e_j)) = \sum_{j=1}^{l-1} p^v(w_j)$. Since we assumed that $p^v(w_l) \geq \prod_{j=1}^{l-1}(1 - p(e_j))$, it follows that $\sum_{j=1}^{l} p^v(w_j) \geq 1$, in contradiction to $p^v(w_{l+1}) > 0$.

We showed that $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{cie}\}} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{cie}\}}$. For proving $\mathsf{PrXML}^{\{\mathsf{cie}\}} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{cie}\}}_{|\not\perp}$, we use the following transformation. If u is a cie node that has a cie child v, then we remove v, connect each child w_i of v directly to u and define $\alpha^u(w_i) = \alpha^u(v) \wedge \alpha^v(w_i)$. □

By Proposition 5.7 and Theorem 5.13, every family of p-documents is v-translatable to $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. However, this particular translation creates a p-document that is linear in the combined size of all the possible worlds. So, when this v-translation is from $\mathsf{PrXML}^{\{\mathsf{exp}\}}$, it involves an exponential blowup. Whether $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ can be efficiently v-translated to $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ is an open problem. In any case, the following lemma shows that under the object-based semantics, $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ does not even generalize $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\perp}$. That is, $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\perp}$ is not o-translatable to $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ (and, by Theorem 5.13, neither to $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{cie}\}}$).

Lemma 5.14 $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\perp} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{cie}\}}$.

Proof By Theorem 5.13, it suffices to show that there is a p-document $\tilde{\mathscr{P}}' \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\perp}$, such that no $\tilde{\mathscr{P}}$ in $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ satisfies $\tilde{\mathscr{P}}' \equiv_o \tilde{\mathscr{P}}$. The existence of $\tilde{\mathscr{P}}'$ is a consequence of the following inequality that holds for all p-documents $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{cie}\}}$ and all ordinary nodes w_1 and w_2 of $\tilde{\mathscr{P}}$, such that w_1 and w_2 appear together in at least one document d of $pwd(\tilde{\mathscr{P}})$.

$$\Pr(w_1, w_2 \in \mathscr{V}(\mathscr{P})) \geq \Pr(w_1 \in \mathscr{V}(\mathscr{P})) \cdot \Pr(w_2 \in \mathscr{V}(\mathscr{P})) \quad (5.1)$$

That is, the probability that both nodes exist in a possible world is at least as high as the product of the probabilities that each one exists. Clearly, there is a p-document in $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\perp}$ that violates this inequality, e.g., the one depicted in Figure 5.4(c). We prove the above inequality by showing how to calculate the probability of the event "a possible world of $\tilde{\mathscr{P}}$ contains a set of ordinary nodes U."

Let U be a set of ordinary nodes of $\tilde{\mathscr{P}}$, such that all the nodes of U appear together in at least one document of $pwd(\tilde{\mathscr{P}})$. Consider the minimal r-subtree $p(U)$ of $\tilde{\mathscr{P}}$ that contains all the nodes of U. Let $A(U)$ be the set of all the literals (i.e., events or negated events) that appear in the conjunctions $\alpha^v(w)$, where w is a node of $p(U)$. $A(U)$ does not contain both an event e and its negation $\neg e$, because $pwd(\tilde{\mathscr{P}})$ has a document that contains all the nodes of U. Therefore, the probability that all the nodes of U appear in a random document is the product of the probabilities that the literals of $A(U)$ are **true**. Hence, the inequality follows because $A(\{w_1, w_2\}) = A(\{w_1\}) \cup A(\{w_2\})$. □

We now discuss whether $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ generalizes the family $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. Proposition 5.7 and the first part of Theorem 5.12 imply that $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ is v-translatable to $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\perp}$. But this is not an efficient v-translation. The next theorem shows that an efficient v-translation does not exist. Moreover, regardless of efficiency, there is no o-translation.

Theorem 5.15 *The following hold.*

1. $\mathsf{PrXML}^{\{\mathsf{cie}\}} \not\sqsubseteq_v^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}}$.
2. $\mathsf{PrXML}^{\{\mathsf{cie}\}} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}}$.

Proof We use the same proof for both parts. For all $n > 2$, let $\tilde{\mathscr{P}}_n$ be the p-document of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ depicted in Figure 5.5. $\tilde{\mathscr{P}}_n$ has $2n + 3$ ordinary nodes

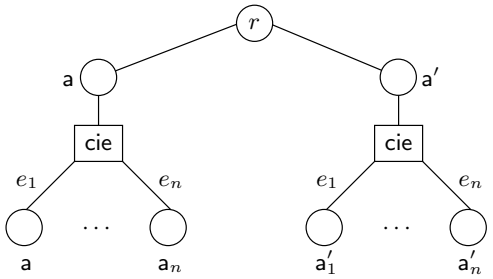

Fig. 5.5 A p-document of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ that can be neither efficiently v-translated nor o-translated to $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}}$

and n events $e_1, \ldots, e_n$, each with probability 1/2. The root r has two ordinary children labeled with a and a'. In addition, each of the two nodes labeled with a and a' has n ordinary grandchildren labeled with $\mathsf{a}_1, \ldots, \mathsf{a}_n$ and $\mathsf{a}'_1, \ldots, \mathsf{a}'_n$, respectively. Note that if the event e_i is **true**, then the two nodes labeled with a_i and a'_i are chosen; conversely, if e_i is **false**, then none of these two nodes is chosen.

For $n > 2$, let $\tilde{\mathscr{P}}'_n$ be a minimal p-document of $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}}$, such that $\tilde{\mathscr{P}}'_n \equiv_v \tilde{\mathscr{P}}_n$. We will show that $\tilde{\mathscr{P}}'_n$ has at least 2^n ordinary nodes, thereby proving that $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ is neither efficiently v-translatable nor o-translatable to the family $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{exp}\}}$.

First, we show that all the distributional nodes of $\tilde{\mathscr{P}}'_n$ appear in a hierarchy immediately below the root. That is, no distributional node is a descendant of an ordinary node that is labeled with either a or a'. Suppose that this is not so. Namely, there is a distributional node v that is a descendant of an ordinary node u that is labeled with a (the symmetric case where u is labeled with a' is handled similarly).

If in all the possible worlds that contain u, the set of labels appearing in the children of u is the same, then v (possibly with some other nodes) can be eliminated while preserving v-equivalence, contradicting the assumption that $\tilde{\mathscr{P}}'_n$ is minimal. (Note that this argument includes the case where no possible world contains u.) Hence, there is a label a_j and two possible worlds d_1 and d_2, such that the following holds. Both d_1 and d_2 contain u, but only in d_1 does the label a_j appear among the children of u.

Every possible world of $\tilde{\mathscr{P}}_n$ that includes the label a_j also has the label a'_j. Hence, d_1 has a node u'_j that is labeled with a'_j. It follows that in $\tilde{\mathscr{P}}'_n$, the least common ancestor of u and u'_j must be a proper ancestor of u, because $\tilde{\mathscr{P}}_n$ has no node labeled with a'_j that appears as a descendant of a node labeled with a.

Let s_1 and s_2 be two r-subtrees of $\tilde{\mathscr{P}}'_n$ such that $\mathsf{doc}(s_1) = d_1$ and $\mathsf{doc}(s_2) = d_2$. We construct an r-subtree s of $\tilde{\mathscr{P}}'_n$ as follows. Distributional nodes that are not descendants of u choose children as in s_1, whereas the descendants of u choose their children as in s_2. Note that distinct distributional nodes of $\tilde{\mathscr{P}}'_n$ choose their children independently of one another, because none of them is of type cie. Hence, the resulting random document $d = \mathsf{doc}(s)$ has a nonzero probability. Clearly, d has the label a'_j but not the label a_j, contradicting $\tilde{\mathscr{P}}'_n \equiv_v \tilde{\mathscr{P}}_n$.

This contradiction proves that all the distributional nodes of $\tilde{\mathscr{P}}'_n$ appear above all the nodes labeled with either a or a', that is, in a hierarchy immediately below the root. It thus follows that for all possible worlds d of $\tilde{\mathscr{P}}_n$, the subtree rooted at the node labeled with a must appear *as is* in $\tilde{\mathscr{P}}'_n$. But there are 2^n different possible worlds, yielding 2^n such subtrees. Therefore, $\tilde{\mathscr{P}}'_n$ has more than 2^n ordinary nodes. □

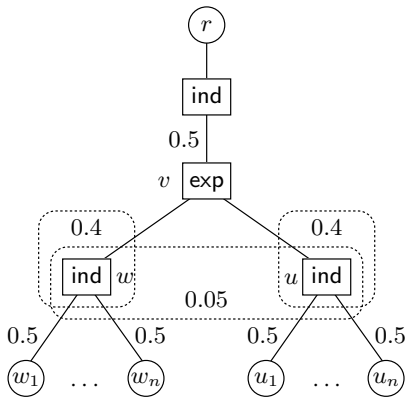

Fig. 5.6 A p-document of $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{ind}\}}$ that cannot be efficiently o-translated to $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}_{\not\perp}$

Finally, we consider the expressive power of exp and cie without hierarchies of distributional nodes, namely, $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}_{\not\perp}$. By Theorem 5.13, this family is at least as general as the family $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{cie}\}}$. However, the following theorem shows that it does not generalize $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ (under the object-based semantics).

Theorem 5.16 $\mathsf{PrXML}^{\{\mathsf{exp}\}} \not\sqsubseteq^{\mathrm{poly}}_{o} \mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}_{\not\perp}$.

Proof We assume, by way of contradiction, that there is an efficient o-translator from the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ to $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}_{\not\perp}$. Hence, Lemma 5.11 implies that there is also an efficient o-translator φ from $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{ind}\}}$ to $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}_{\not\perp}$. Let $\tilde{\mathscr{P}}_n$ be the p-document of Figure 5.6. The index n denotes the number of children of each of the nodes w and u. Since the o-translator φ is efficient, we can choose a fixed value for n so that the following holds. For all exp nodes $\hat{v}$ of the p-document $\varphi(\tilde{\mathscr{P}}_n)$, the number of subsets in the specification of $\hat{v}$ is smaller than 2^n.

Observe that the root of $\varphi(\tilde{\mathscr{P}}_n)$ has only distributional nodes as children, because there is a possible world of $\tilde{\mathscr{P}}_n$ that comprises just the root. Since $\varphi(\tilde{\mathscr{P}}_n)$ is DHF, every one of these distributional nodes has only ordinary children. All these children are leaves, because random documents of $\tilde{\mathscr{P}}_n$ have a height of at most one, $\varphi(\tilde{\mathscr{P}}_n)$ has no useless nodes, and a distributional node cannot be a leaf.

If a random document d of $\tilde{\mathscr{P}}_n$ includes w_i, then the probability that it also includes w_j $(i \neq j)$ is 0.5. But the prior probability of including w_j is just $(0.4+0.05)\cdot 0.5^2 = 0.1125$. Therefore, the random variable $\mathscr{P}_n$ has

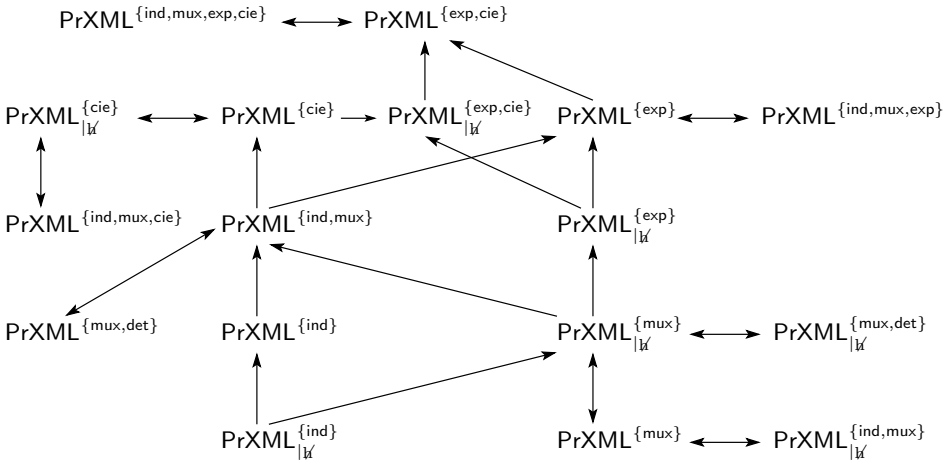

Fig. 5.7 Efficient o-translations between families of p-documents

the following property. For all $1 \leq i < j \leq n$, the two events "$\mathscr{P}_n$ includes node w_i" and "$\mathscr{P}_n$ includes node w_j" are probabilistically dependent. By symmetry, a similar property holds for all u_i and u_j $(i \neq j)$.

Yet another similar property of $\mathscr{P}_n$ is the following. For all $1 \leq i, j \leq n$, the two events "$\mathscr{P}_n$ includes node w_i" and "$\mathscr{P}_n$ includes node u_j" are probabilistically dependent. To see why, observe that the existence of w_i in $\mathscr{P}_n$ decreases the probability of the event[7] "$\mathscr{P}_n$ includes node u_j," because it forces node v of $\tilde{\mathscr{P}}_n$ to choose both w and u (with the low probability 0.05) in order for u_j to be in $\mathscr{P}_n$.

Now, suppose that the root $\hat{r}$ of $\varphi(\tilde{\mathscr{P}}_n)$ has a child y of type exp. If $\hat{r}$ has a second child y', then an ordinary descendant of y and an ordinary descendant of y' are probabilistically independent, in contradiction to the above properties of $\tilde{\mathscr{P}}_n$. Hence, y is the only child of $\hat{r}$. Note that for all subsets S of $\{w_1, \ldots, w_n\} \cup \{u_1, \ldots, u_n\}$, there is a possible world of $\tilde{\mathscr{P}}_n$ with S as the set of leaves. Therefore, the specification of y must include 2^{2n} subsets, which contradicts our choice of n.

It thus follows that $\varphi(\tilde{\mathscr{P}}_n)$ does not contain exp nodes and, therefore, is in $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. Recall that the proof of Lemma 5.14 shows that Equation (5.1) holds for all p-documents $\tilde{\mathscr{P}}$ of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. We now derive a contradiction by showing that the following inequality holds (note that some possible world of $\tilde{\mathscr{P}}_n$ includes both w_1 and u_1).

$$\Pr\left(w_1, u_1 \in \mathscr{V}(\mathscr{P}_n)\right) < \\ < \Pr\left(w_1 \in \mathscr{V}(\mathscr{P}_n)\right) \cdot \Pr\left(u_1 \in \mathscr{V}(\mathscr{P}_n)\right)$$

The left side is $0.05 \cdot 0.5^3 = 0.00625$. Each multiplicand on the right side is $(0.4 + 0.05) \cdot 0.5^2$, so their product is 0.01265625. □

[7] An exact calculation shows that the prior and posterior probabilities of this event are 0.1125 and 1/18, respectively.

It is not known whether $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ is efficiently v-translatable to $\mathsf{PrXML}^{\{\mathsf{exp,cie}\}}_{|\not\ell'}$.

5.5 Overview

Figure 5.7 shows the efficient o-translations that exist between the families of p-documents that have been discussed in this section. This figure is complete in the sense that if there is no directed path from a family $\mathscr{F}_1$ to $\mathscr{F}_2$, then there is no efficient o-translator from $\mathscr{F}_1$ to $\mathscr{F}_2$. As shown in Figure 5.7, the family $\mathsf{PrXML}^{\{\mathsf{exp,cie}\}}$ is the most general, and the families $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ and $\mathsf{PrXML}^{\{\mathsf{exp,cie}\}}_{|\not\ell'}$ are just below $\mathsf{PrXML}^{\{\mathsf{exp,cie}\}}$. The family $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ is right below $\mathsf{PrXML}^{\{\mathsf{exp,cie}\}}_{|\not\ell'}$.

Recall that an o-translation is also a v-translation. In addition, we have shown that every family $\mathscr{F}$ of p-documents is v-translatable to $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$ and, hence, also to $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ and $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. However, the family $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ is not efficiently v-translatable to $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$, or even to $\mathsf{PrXML}^{\{\mathsf{ind,mux,exp}\}}$; namely, such a v-translation causes an exponential blowup.

We also considered the need for hierarchies of distributional nodes. We showed that such hierarchies are not required in the case of either $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ or $\mathsf{PrXML}^{\{\mathsf{mux}\}}$. However, for the families $\mathsf{PrXML}^{\{\mathsf{ind}\}}$, $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}}$, $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$ and $\mathsf{PrXML}^{\{\mathsf{exp}\}}$, these hierarchies properly increase the expressive power, in the sense that there are no efficient v- or o-translations that can eliminate them (note that in some of these cases, there are no translations regardless of efficiency). In the case of $\mathsf{PrXML}^{\{\mathsf{exp,cie}\}}$, we only proved that there is no efficient o-translation that eliminates hierarchies (and for v-translation, it is open).

For the families of p-documents considered thus far, the results of this section determine for every pair $\mathscr{F}_1$

and $\mathscr{F}_2$ whether or not there is an efficient o-translation from $\mathscr{F}_1$ to $\mathscr{F}_2$. When there is no efficient o-translation, then it is often the case that there is no efficient v-translation as well (although an inefficient v-translation usually exists). However, in some cases, the existence of an efficient v-translation is left as an open problem. The main unsolved question is whether the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ (or even $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\mathscr{U}}$) can be efficiently v-translated to $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$, or even to $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. In the next section, we partially solve this problem by showing the existence of an efficient v-translation (but no o-translations) from $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ to $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$ under the assumption of a fixed upper bound on the out-degree of exp nodes (or on the maximal number of distributional nodes on any path from the root to a leaf).

6 Distributional Nodes with Bounded Degrees

In this section, we restrict families of p-documents by imposing a bound on the number of children that a distributional node may have. We study the effect of this bound on the expressive power. The combination of this restriction with a lack of distributional hierarchies is beyond the scope of this paper.

Let $\mathscr{F}$ be a family of p-documents and $b \geq 2$ be an integer. We denote by $\mathscr{F}_{\Delta \leq b}$ the subset of $\mathscr{F}$ that comprises all the p-documents $\tilde{\mathscr{P}}$, such that each distributional node $v \in \mathscr{V}^{\mathrm{dst}}(\tilde{\mathscr{P}})$ has b or fewer children. For example, in a p-document $\tilde{\mathscr{P}}$ of $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq 2}$, every distributional node is of type exp and has either one or two children (recall that every distributional node must have at least one child). Note that there is no bound on the number of children of an ordinary node.

The following theorem shows that for the families that do not include the type exp, the bound 2 is enough.

Proposition 6.1 *The following hold.*

1. $\mathsf{PrXML}^{\{\mathsf{ind}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{ind}\}}_{\Delta \leq 2}$.
2. $\mathsf{PrXML}^{\{\mathsf{mux}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{mux}\}}_{\Delta \leq 2}$.
3. $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}_{\Delta \leq 2}$.
4. $\mathsf{PrXML}^{\{\mathsf{cie}\}} \equiv_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{cie}\}}_{\Delta \leq 2}$.

Proof Observe that for each of the four parts, the direction $\sqsupseteq_o^{\mathrm{poly}}$ is trivial. To prove the opposite direction, let $\tilde{\mathscr{P}}$ be a p-document of the family on the left-hand side of one of the four parts. We describe an efficient process that preserves o-equivalence and does the following. Given a distributional node $v \in \mathscr{V}^{\mathrm{dst}}(\tilde{\mathscr{P}})$ that has $k > 2$ children, the process replaces v with three distributional nodes of the same type and degrees 1, 2 and $k-1$. By repeatedly applying this process, we get an o-equivalent p-document, such that each distributional node has one or two children. Note that this is sufficient for proving Parts 1, 2 and 4. For Part 3, we first apply the above process to the given p-document of $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$, and then use the o-translation (into the family $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$) that is described in the proof of Lemma 5.8. Note that this translation does not increase the maximal out-degree of distributional nodes; hence, the end result is a p-document of $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}_{\Delta \leq 2}$.

The process of replacing v is illustrated in Figure 6.1. The children of v are denoted by $w_1, \ldots, w_k$. In the subtree that replaces v, the root is the distributional node v' and it has two distributional children u and u_k. The children of u are $w_1, \ldots, w_{k-1}$, and the only child of u_k is w_k. Note that the nodes $w_1, \ldots, w_k$, as well as the whole subtrees under them, are not changed. Recall that v', u and u_k have the same type as v. In the remainder of the proof, we give additional details of this construction according to the type of v.

v is of type ind. In this case, v' chooses both of its children with probability 1 (as if it is a det node). The probabilities of choosing the children $w_1, \ldots, w_k$ are unchanged, that is, $p^{u_k}(w_k) = p^v(w_k)$ and $p^u(w_i) = p^v(w_i)$ for $1 \leq i \leq k-1$.

v is of type mux. Let $p = p^v(w_k)$. Node v' chooses (the mutually exclusive) u and u_k with probabilities $(1-p)$ and p, respectively. Node u_k chooses w_k with probability 1. Finally, for $1 \leq i \leq k-1$, we set $p^u(w_i)$ to $p^v(w_i)/(1-p)$. Note that $p < 1$ since there are no useless nodes and v has more than one child.

v is of type cie. This case is handled similarly to the case where v is of type ind. In particular, $\alpha^{v'}(u)$ and $\alpha^{v'}(u_k)$ are **true** (i.e., empty conjunctions), $\alpha^{u_k}(w_k) = \alpha^v(w_k)$ and $\alpha^u(w_i) = \alpha^v(w_i)$ for $1 \leq i \leq k-1$. □

Next, we consider the type exp. If $b_2 \geq b_1 \geq 2$, then $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_1} \sqsubseteq_o^{\mathrm{poly}} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_2}$ trivially holds. The next lemma shows that the opposite direction does not hold. That is, Proposition 6.1 cannot be extended to the type exp and, moreover, raising the bound b increases the expressive power of $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b}$ under the o-semantics.

Lemma 6.2 $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b+1} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b}$ *holds for all* $b \geq 2$.

Proof Consider the p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b+1}$ that is depicted in Figure 6.2. The root r of $\tilde{\mathscr{P}}$ has a single child v which is an exp node, and v has $b+1$ ordinary children $w_1, \ldots, w_{b+1}$. Let $W = \{w_1, \ldots, w_{b+1}\}$. For all $1 \leq i \leq b+1$, node v specifies the probability $1/(b+1)$ for the set $W \setminus \{w_i\}$, that is, $p^v(W \setminus \{w_i\}) = \frac{1}{b+1}$.

For $i \neq j$, the events "$\mathscr{P}$ does not include w_i" and "$\mathscr{P}$ does not include w_j" are probabilistically depen-

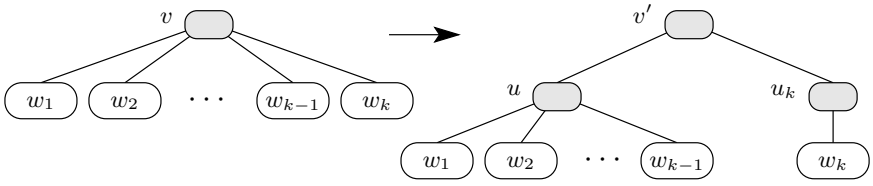

Fig. 6.1 Reducing the degree of a distributional node

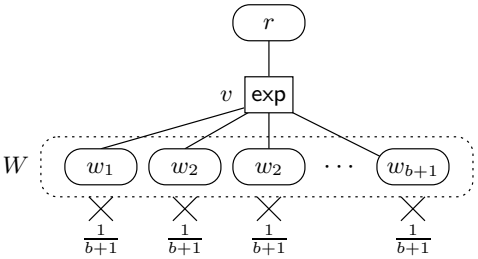

Fig. 6.2 A p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b+1}$ that cannot be o-translated to $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b}$

dent. To see why, observe that w_i must appear in $\mathscr{P}$ if w_j is absent. By using this property, we will show that no p-document of $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b}$ is o-equivalent to $\tilde{\mathscr{P}}$.

Suppose, by way of contradiction, that a p-document $\tilde{\mathscr{P}}_0 \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b}$ is o-equivalent to $\tilde{\mathscr{P}}$. It is easy to show that r is the root of $\tilde{\mathscr{P}}_0$ and W comprises exactly all the leaves of $\tilde{\mathscr{P}}_0$ (since $\tilde{\mathscr{P}}_0$ has no useless nodes). We will prove that there is a probability greater than 0 that all the nodes of W appear in the random document $\mathscr{P}_0$, thereby deriving a contradiction to the o-equivalence of $\tilde{\mathscr{P}}_0$ and $\tilde{\mathscr{P}}$, because the probability that $\mathscr{P}$ contains all of W is 0.

Let u_a be the least common ancestor of W in $\tilde{\mathscr{P}}_0$. Note that u_a has at least two children. If u_a is the root r, then there are two distinct leaves w_i and w_j in W, such that each one is a descendant of a different child of r. Hence, the events "$\mathscr{P}_0$ does not include w_i" and "$\mathscr{P}_0$ does not include w_j" are probabilistically independent, which is the opposite of the above property of $\mathscr{P}$. Therefore, $\tilde{\mathscr{P}}_0 \equiv_o \tilde{\mathscr{P}}$ implies that u_a is not r, so u_a is a distributional node. Consequently, u_a has at most b children. Since all the $b+1$ nodes of W are descendants of u_a, there is a child u_c of u_a that has two or more descendants that are in W. Let $W_c \subseteq W$ be the set of ordinary descendants of u_c. Since u_a has more than one child, u_c has at most b descendants from W. It follows that $2 \leq |W_c| \leq b$. By the definition of $\tilde{\mathscr{P}}$, the probability that $\mathscr{P}$ contains all the nodes of W_c (and, possibly, additional nodes of W) is greater than 0. Consequently, the probability that $\mathscr{P}_0$ contains W_c is greater than 0, because $\tilde{\mathscr{P}}_0 \equiv_o \tilde{\mathscr{P}}$. Note that for $\mathscr{P}_0$ to contain W_c, the r-subtree $\mathscr{P}_0^\Sigma$ must contain W_c and u_c. We conclude the following.

$$\begin{aligned}0 &< \Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) = \Pr\left(W_c \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) = \\ &= \Pr\left(u_c \in \mathscr{V}(\mathscr{P}_0^\Sigma)\right) \times \\ &\quad \times \Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma) \mid u_c \in \mathscr{V}(\mathscr{P}_0^\Sigma)\right)\end{aligned}$$

In particular, the following holds.

$$\Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma) \mid u_c \in \mathscr{V}(\mathscr{P}_0^\Sigma)\right) > 0 \tag{6.1}$$

We arbitrarily choose $w_c \in W_c$, and denote by $\overline{W_c}$ the set $(W \setminus W_c)$. Observe that $\overline{W_c} \cup \{w_c\}$ has at most b nodes. So, again, the probability that $\mathscr{P}_0^\Sigma$ contains $\overline{W_c} \cup \{w_c\}$ is greater than 0 and, since w_c is a descendant of u_c, the r-subtree $\mathscr{P}_0^\Sigma$ must contain u_c in order to contain w_c. Thus, the following holds.

$$\begin{aligned}0 &< \Pr\left(\overline{W_c} \cup \{w_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) \leq \\ &\leq \Pr\left(\overline{W_c} \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right)\end{aligned} \tag{6.2}$$

We now consider the probability that $\mathscr{P}_0$ (or, equivalently, that $\mathscr{P}_0^\Sigma$) contains all the nodes of W.

$$\begin{aligned}&\Pr\left(W \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) = \Pr\left(W \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) = \\ &= \Pr\left(\overline{W_c} \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) \times \\ &\quad \times \Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma) \mid \overline{W_c} \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right)\end{aligned} \tag{6.3}$$

The random process of constructing $\mathscr{P}_0^\Sigma$ and the fact that $\tilde{\mathscr{P}}_0$ has no cie nodes imply the following. Given the condition that $\mathscr{P}_0^\Sigma$ includes u_c, the events "$\mathscr{P}_0^\Sigma$ contains W_c" and "$\mathscr{P}_0^\Sigma$ contains $\overline{W_c}$" are probabilistically independent. In particular, the following holds.

$$\begin{aligned}&\Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma) \mid \overline{W_c} \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) = \\ &= \Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma) \mid u_c \in \mathscr{V}(\mathscr{P}_0^\Sigma)\right)\end{aligned} \tag{6.4}$$

From Equations (6.1), (6.2), (6.3) and (6.4), we conclude the following.

$$\begin{aligned}&\Pr\left(W \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) = \Pr\left(\overline{W_c} \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) \times \\ &\quad \times \Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma) \mid \overline{W_c} \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) = \\ &= \Pr\left(\overline{W_c} \cup \{u_c\} \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma)\right) \times \\ &\quad \times \Pr\left(W_c \subseteq \mathscr{V}(\mathscr{P}_0^\Sigma) \mid u_c \in \mathscr{V}(\mathscr{P}_0^\Sigma)\right) > 0\end{aligned}$$

As explained above, this yields a contradiction. □

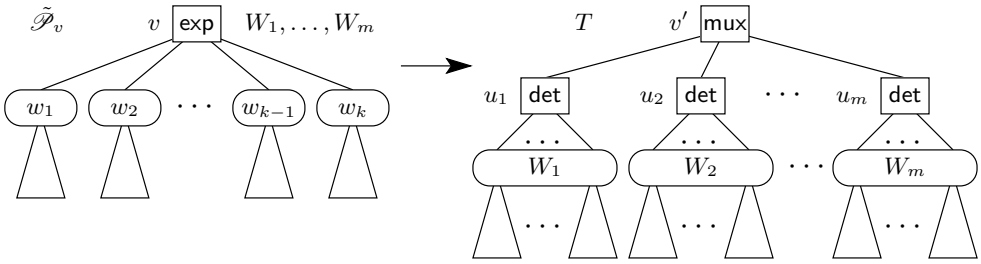

Fig. 6.3 Simple replacement

6.1 Efficient v-Translation from PrXML$^{\{\text{exp}\}}$ to PrXML$^{\{\text{mux,det}\}}$

By Proposition 5.7, every p-document of PrXML$^{\{\text{exp}\}}$ can be v-translated to PrXML$^{\{\text{mux,det}\}}$, but the size of the result is linear in the number of possible worlds of $\tilde{\mathscr{P}}$. In this section, we present two v-translations that are efficient for important sub-classes of PrXML$^{\{\text{exp}\}}$.

SimpleTrans and GreedyTrans are two v-translations that traverse the p-document top down and operate as follows. Whenever an exp node v is visited, the subtree of $\tilde{\mathscr{P}}$ that is rooted at v is replaced with a different subtree that has a root of type mux. In SimpleTrans, this operation is called *simple replacement*, and in GreedyTrans, it is called *greedy replacement*. The details of these replacements are described below.

Consider a p-document $\tilde{\mathscr{P}} \in$ PrXML$^{\{\text{exp}\}}$. Let v be an exp node of $\tilde{\mathscr{P}}$ with the set of children $W = \{w_1, \ldots, w_k\}$. Suppose that v specifies nonzero probabilities for the subsets $W_1, \ldots, W_m$ of W. Note that $\sum_{j=1}^{m} p^v(W_j) = 1$. By $\tilde{\mathscr{P}}_v$ we denote the subtree of $\tilde{\mathscr{P}}$ rooted at v and consisting of all the descendants of v. Similarly, for $1 \leq i \leq k$, the subtree $\tilde{\mathscr{P}}_i$ of $\tilde{\mathscr{P}}$ is the one rooted at w_i and comprising all the descendants of w_i.

The simple replacement is illustrated in Figure 6.3. It replaces $\tilde{\mathscr{P}}_v$ with the tree T that is constructed as follows. The root of T is a mux node that has m det children $u_1, \ldots, u_m$. For all $1 \leq j \leq m$ and $w_i \in W_j$, we create a copy of $\tilde{\mathscr{P}}_i$ and make it a subtree of u_j.

The greedy replacement is more complicated. It is illustrated in Figure 6.4 and defined as follows. First, we choose the node w_i of W that has the maximal number of descendants (i.e., $\tilde{\mathscr{P}}_i$ has the maximal number of nodes among $\tilde{\mathscr{P}}_1, \ldots, \tilde{\mathscr{P}}_k$). By renaming if necessary, we assume that this node is w_k. The sets $\mathscr{W}^k$ and $\mathscr{W}^{\neg k}$, and the number p_k are defined as follows.

$$\mathscr{W}^k \stackrel{\text{def}}{=} \{W_j \setminus \{w_k\} \mid 1 \leq j \leq m \wedge w_k \in W_j\}$$

$$\mathscr{W}^{\neg k} \stackrel{\text{def}}{=} \{W_j \mid 1 \leq j \leq m \wedge w_k \notin W_j\}$$

$$p_k \stackrel{\text{def}}{=} 1 - \sum_{W_j \in \mathscr{W}^{\neg k}} p^v(W_j)$$

In other words, $\mathscr{W}^k$ comprises all the sets W', such that $w_k \notin W'$ and $W' \cup \{w_k\}$ is given a nonzero probability by v; $\mathscr{W}^{\neg k}$ is the set of all the W_j that do not include w_k; and p_k is the probability that v chooses w_k (possibly in addition to other nodes). The tree $\tilde{\mathscr{P}}_v$ is replaced with the tree T that consists of four new distributional nodes v', u', u^k and $u^{\neg k}$, as well as copies of $\tilde{\mathscr{P}}_1, \ldots, \tilde{\mathscr{P}}_k$. Note that u^k and $u^{\neg k}$ are of type exp and they will be handled by GreedyTrans in due course. The full details are given below.

The root of T is the mux node v'. The children of v' are u' and $u^{\neg k}$, and they are chosen with probabilities p_k and $1 - p_k$, respectively. The type of u' is det and one of its two children is u^k.

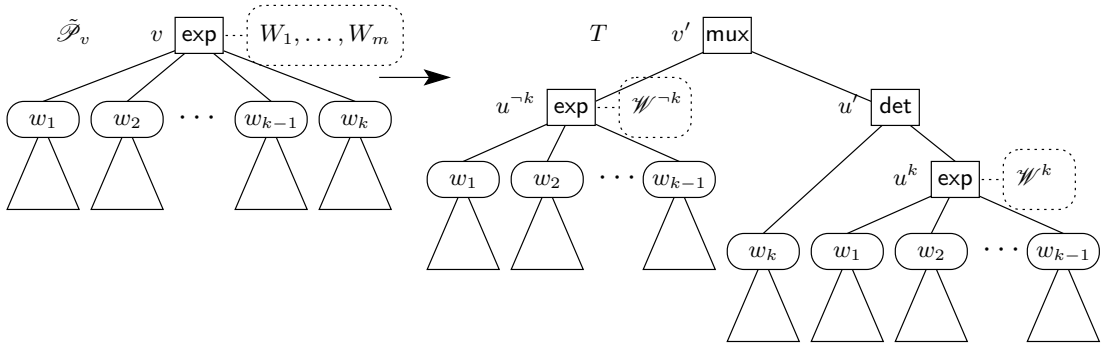

Fig. 6.4 Greedy replacement

For $1 \leq i \leq k-1$, one copy of $\tilde{\mathscr{P}}_i$ becomes a subtree of u^k and a second copy—a subtree of $u^{\neg k}$. $\tilde{\mathscr{P}}_k$ becomes a subtree of u'.

As mentioned above, the type of both u^k and $u^{\neg k}$ is exp. Node $u^{\neg k}$ specifies the probability $p^v(W_j)/(1-p_k)$ for each $W_j \in \mathscr{W}^{\neg k}$. Node u^k specifies the probability $p^v(W' \cup \{w_k\})/p_k$ for each subset $W' \in \mathscr{W}^k$.

An exception to the above construction is when $k = 1$ or $p_k = 1$ (note that $p_k > 0$, because there are no useless nodes). An exp node with a single child is actually a mux node, so GreedyTrans does nothing at node v if $k = 1$. If $p_k = 1$, then $u^{\neg k}$ and its descendants are not added to T.

The *distributional depth* of a p-document $\tilde{\mathscr{P}}$ is defined as the maximal number of distributional nodes along any path from the root to a leaf. The next proposition shows that SimpleTrans is efficient if the distributional depth of $\tilde{\mathscr{P}}$ is bounded by a constant. Note that the number of possible worlds can still be exponential in the size of $\tilde{\mathscr{P}}$ even if this bound is 2. Formally, for a natural number h, we denote by $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\downarrow \leq h}$ the set of all p-documents $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}$, such that the distributional depth of $\tilde{\mathscr{P}}$ is at most h.

Proposition 6.3 *Let $h \geq 0$ be a constant. The algorithm* SimpleTrans *is an efficient v-translation from the family* $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\downarrow \leq h}$ *to* $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$. *Hence, we have that* $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\downarrow \leq h} \sqsubseteq^{\mathrm{poly}}_{v} \mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$.

Proof Consider a p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\downarrow \leq h}$ having N nodes. Let M be the smallest integer, such that for all distributional nodes v of $\tilde{\mathscr{P}}$, there are at most M subsets in the specification of v. Clearly, both N and M are not larger than the size of $\tilde{\mathscr{P}}$.

We extend earlier notation so that $\tilde{\mathscr{P}}_v$, as well as the subtrees $\tilde{\mathscr{P}}_1, \ldots, \tilde{\mathscr{P}}_k$ rooted at the children of v, are defined for all the nodes of $\tilde{\mathscr{P}}$ (rather than just exp nodes). In addition, let N_v and N_i be the numbers of nodes of the subtrees $\tilde{\mathscr{P}}_v$ and $\tilde{\mathscr{P}}_i$, respectively. $G(\tilde{\mathscr{P}}_v)$ denotes the number of nodes in the result of applying SimpleTrans to $\tilde{\mathscr{P}}_v$.

We prove the following claim by a bottom-up induction on $\tilde{\mathscr{P}}$: If the distributional depth of the subtree $\tilde{\mathscr{P}}_v$ is bounded by c, then $G(\tilde{\mathscr{P}}_v) \leq (M+1)^c N_v$. This claim implies that $G(\tilde{\mathscr{P}}) \leq (M+1)^h N$, thereby proving the proposition.

For the basis of the induction, the subtree $\tilde{\mathscr{P}}_v$ is just a leaf (and hence v is an ordinary node). So, SimpleTrans does not change $\tilde{\mathscr{P}}_v$ and, consequently, the induction hypothesis holds, because $1 \leq (M+1)^0$.

For the inductive step, there are two cases to consider. First, if v is an ordinary node, then SimpleTrans does not change v. Hence, by applying the induction hypothesis to $\tilde{\mathscr{P}}_1, \ldots, \tilde{\mathscr{P}}_k$, we get the following.

$$\begin{aligned} G(\tilde{\mathscr{P}}_v) =& 1 + \sum_{i=1}^{k} G(\tilde{\mathscr{P}}_i) \leq \\ & \leq 1 + \sum_{i=1}^{k} (M+1)^c N_i \leq \\ & \leq 1 + (M+1)^c \sum_{i=1}^{k} N_i \leq (M+1)^c N_v \end{aligned}$$

The second line, in the above equation, follows from the induction hypothesis. The last inequality follows from $1 + \sum_{i=1} N_i = N_v$.

If v is a distributional node, then SimpleTrans replaces v with at most $1 + M$ nodes and replicates each $\tilde{\mathscr{P}}_i$ at most M times. In the equation below, the second line follows from the induction hypothesis (note that the distributional depth of each $\tilde{\mathscr{P}}_i$ is at most $c-1$).

$$\begin{aligned} G(\tilde{\mathscr{P}}_v) & \leq 1 + M + M \sum_{i=1}^{k} G(\tilde{\mathscr{P}}_i) \leq \\ & \leq 1 + M + M \sum_{i=1}^{k} (M+1)^{c-1} N_i \leq \\ & \leq (M+1)^c (1 + \sum_{i=1}^{k} N_i) = (M+1)^c N_v \quad \square \end{aligned}$$

Next, we show that SimpleTrans is not an efficient v-translation from $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq 2}$. In proof, for all $n > 0$, let $\tilde{\mathscr{P}}^n$ be the p-document shown in Figure 6.5. When applying SimpleTrans to $\tilde{\mathscr{P}}^n$, the resulting document has a depth of $2n+1$. It can be easily verified that the number of mux nodes at depth $2n-1$ of the result is

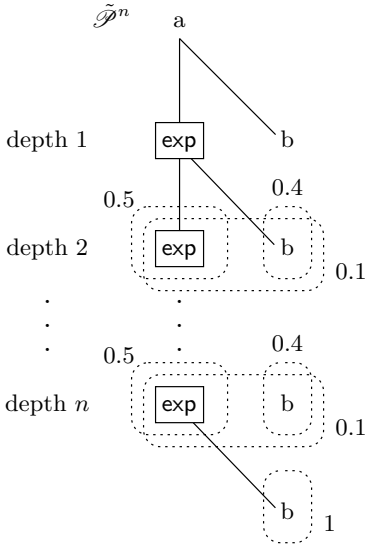

Fig. 6.5 An example of a series of p-documents of $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq 2}$ over which SimpleTrans results in an exponential blowup

2^{n-1}. As opposed to SimpleTrans, the following lemma shows that GreedyTrans is an efficient v-translation from $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b}$ to $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$, for all $b \geq 2$.

Theorem 6.4 *Let $b \geq 2$ be a constant.* GreedyTrans *is an efficient v-translation from the family* $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b}$ *to* $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$.

Proof In this proof, $G(\tilde{\mathscr{P}})$ denotes the number of nodes in the result of applying GreedyTrans to $\tilde{\mathscr{P}}$. Clearly, GreedyTrans does not introduce exp nodes with specifications that are larger than the maximal specification in the source document. So, it suffices to prove that the following holds for all p-documents $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b}$. For all subtrees $\tilde{\mathscr{P}}_v$ of $\tilde{\mathscr{P}}$, it holds that $G(\tilde{\mathscr{P}}_v)$ is polynomial in $|\mathscr{V}(\tilde{\mathscr{P}}_v)|$ (i.e., the number of nodes of $\tilde{\mathscr{P}}_v$). Let $c > 1$ be the smallest (fixed) integer, such that $(1-\frac{1}{b+1})^{c-1} \leq 1/2$. We prove by induction on $|\mathscr{V}(\tilde{\mathscr{P}}_v)|$ that $G(\tilde{\mathscr{P}}_v) \leq 2|\mathscr{V}(\tilde{\mathscr{P}}_v)|^c - 1$ for all subtrees $\tilde{\mathscr{P}}_v$.

For the basis of the induction, we assume that v is a leaf. In this case, GreedyTrans does not change $\tilde{\mathscr{P}}_v$, and hence, $G(\tilde{\mathscr{P}}_v) = 1$, as required.

For the inductive step, we consider a node v of some $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b}$. Recall that $\tilde{\mathscr{P}}_1 \ldots, \tilde{\mathscr{P}}_k$ denote the subtrees rooted at the children of $\tilde{\mathscr{P}}_v$, and $\tilde{\mathscr{P}}_k$ has the maximal number of nodes among these subtrees.

There are two cases to consider. First, if v is an ordinary node, then the following equation holds, where the second line follows from the induction hypothesis.

$$\begin{aligned}
G(\tilde{\mathscr{P}}_v) &\leq 1 + \sum_{i=1}^{k} G(\tilde{\mathscr{P}}_i) \leq \\
&\leq 1 + \sum_{i=1}^{k} (2|\mathscr{V}(\tilde{\mathscr{P}}_i)|^c - 1) \leq \\
&\leq 2\sum_{i=1}^{k} |\mathscr{V}(\tilde{\mathscr{P}}_i)|^c \leq \\
&\leq -1 + 2(1 + \sum_{i=1}^{k} |\mathscr{V}(\tilde{\mathscr{P}}_i)|^c) \leq \\
&\leq -1 + 2(1 + \sum_{i=1}^{k} |\mathscr{V}(\tilde{\mathscr{P}}_i)|)^c = \\
&= 2|\mathscr{V}(\tilde{\mathscr{P}}_v)|^c - 1
\end{aligned}$$

In the second case, v is an exp node, and we apply the greedy replacement as illustrated in Figure 6.4. If $k = 1$, then nothing is done at node v, so the proof is the same as in the case where v is an ordinary node. If $k > 1$, then after applying GreedyTrans to v, we continue recursively with the subtrees $\tilde{\mathscr{P}}_{u^{\neg k}}$, $\tilde{\mathscr{P}}_{u^k}$ and $\tilde{\mathscr{P}}_k$ (the transformation does nothing when visiting the det node u'). Each of these three subtrees has fewer nodes than $\tilde{\mathscr{P}}_v$, so the induction hypothesis implies the following.

$$G(\tilde{\mathscr{P}}_v) \leq 2 + G(\tilde{\mathscr{P}}_{u^{\neg k}}) + G(\tilde{\mathscr{P}}_{u^k}) + G(\tilde{\mathscr{P}}_k) \tag{6.5}$$

Note that if $p_k = 1$, then $G(\tilde{\mathscr{P}}_{u^{\neg k}}) = 0$. The proof below holds also in this case.

Let $N = |\mathscr{V}(\tilde{\mathscr{P}}_v)|$ and $N_i = |\mathscr{V}(\tilde{\mathscr{P}}_i)|$. Note that $N = 1 + \sum_{i=1}^{k} N_i$. Since v has k children, it follows that $k \leq b$ and the sum $1 + \sum_{i=1}^{k} N_i$ has at most $b+1$ operands. We have assumed that $\tilde{\mathscr{P}}_k$ has the largest number of nodes among $\tilde{\mathscr{P}}_i, \ldots, \tilde{\mathscr{P}}_k$. Therefore, $N_k \geq N/(b+1)$.

Let r_k be the ratio N_k/N. Then $r_k \geq 1/(b+1)$. For each of $\tilde{\mathscr{P}}_{u^k}$ and $\tilde{\mathscr{P}}_{u^{\neg k}}$, the number of nodes is at most $N - N_k = (1-r_k)N$. The number of nodes of $\tilde{\mathscr{P}}_k$ is $r_k N$. We now continue with Equation (6.5) and get the following. Note that the fifth inequality below uses $r_k \geq 1/(b+1)$, which was shown above.

$$\begin{aligned}
G(\tilde{\mathscr{P}}_v) &\leq 2 + G(\tilde{\mathscr{P}}_{u^{\neg k}}) + G(\tilde{\mathscr{P}}_{u^k}) + G(\tilde{\mathscr{P}}_k) \leq \\
&\leq 2 + 2 \cdot (2((1-r_k)N)^c - 1) + 2(r_k N)^c - 1 \leq \\
&\leq -1 + 2 \cdot 2(1-r_k)^c N^c + 2r_k^c N^c \leq \\
&\leq -1 + 2 \cdot 2(1-r_k)^c N^c + 2r_k N^c \leq \\
&\leq -1 + 2 \cdot 2\left(1 - \frac{1}{b+1}\right)^{c-1} (1-r_k)N^c + \\
&\quad + 2r_k N^c
\end{aligned}$$

Recall that c satisfies $\left(1-\frac{1}{b+1}\right)^{c-1} \leq 1/2$. Therefore,

$$G(\tilde{\mathscr{P}}_v) \leq 2(1-r_k)N^c + 2r_k N^c - 1 = 2N^c - 1\,,$$

as required. □

As a result, we get the following corollary.

Corollary 6.5 $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b} \sqsubseteq^{\mathrm{poly}}_{v} \mathsf{PrXML}^{\{\mathsf{mux,det}\}}$ *for all constants $b \geq 2$.*

We conclude with the following theorem.

Theorem 6.6 *For all constants $b_2 > b_1 \geq 2$, the following hold.*

1. $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}} \sqsubseteq^{\mathrm{poly}}_{o} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b_1} \sqsubseteq^{\mathrm{poly}}_{o} \sqsubseteq^{\mathrm{poly}}_{o} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b_2} \sqsubseteq^{\mathrm{poly}}_{o} \mathsf{PrXML}^{\{\mathsf{exp}\}}$.
2. $\mathsf{PrXML}^{\{\mathsf{exp}\}} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b_2} \not\sqsubseteq_o \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b_1} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{ind,mux}\}}$.
3. $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}} \equiv^{\mathrm{poly}}_{v} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b_1} \equiv^{\mathrm{poly}}_{v} \equiv^{\mathrm{poly}}_{v} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta\leq b_2}$.

Proof For Part 1, $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}} \sqsubseteq_o^{\text{poly}} \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_1}$ follows from Part 3 of Proposition 6.1 and the fact that mux and det nodes can be viewed as special cases of exp nodes. The rest of the o-translations are trivial.

For Part 2, Lemma 6.2 implies both $\mathsf{PrXML}^{\{\mathsf{exp}\}} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_2}$ and $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_2} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_1}$. As for the third result of Part 2, it is a consequence of the following observation. In the proof of Lemma 5.14, we showed an example of a p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{exp}\}}$ that cannot be o-translated to $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux},\mathsf{cie}\}}$ (see Figure 5.4(c)), and $\tilde{\mathscr{P}}$ belongs to $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq 2}$. Therefore, $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\Delta \leq b_1} \not\sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$.

Finally, Part 3 follows from Corollary 6.5, Part 3 of Proposition 6.1 and Lemma 5.8. □

7 Probabilistic Updates

Another perspective on the expressiveness of probabilistic XML models is to consider their ability for capturing updates. A main question is whether the result of an update is expressible in some model. Another issue is how complex it is to compute a representation of this result. As is natural in the context of probabilistic data, we consider *probabilistic* updates, that are conditioned by a certain *confidence* in the operation. Typically, a probabilistic database could be the result of a number of successive probabilistic updates on an initial ordinary document.

As with o-translations and v-translations, we consider here two kinds of probabilistic updates: *o-updates*, based on object identity, and *v-updates*, based on value equality. For simplicity, we consider here only *elementary* updates, that is, updates consisting of a single insertion or deletion. The extension to arbitrary updates is not too involved, and is discussed in [8].

7.1 Object-Based Updates

In a real-life system, o-updates are obtained for instance when a user clicks on a node to attach an annotation to it or to delete it. Such an update is thus directly specified on an object. The system may, for instance, attach a confidence to that update depending on the expertise of the particular user. More formally:

Definition 7.1 A *probabilistic o-update operation* is a pair $\tau = (o, c)$ where $0 < c \leq 1$ is the *confidence* in the operation, and o is either:

1. an *o-insertion*, that is, an expression $\iota(v, F)$ where v is a node identifier and F is a document forest;
2. an *o-deletion*, that is, an expression $\delta(v)$ where v is a node identifier.

If $c = 1$, τ is said to be *deterministic*.

In the following, we assume that all nodes to be inserted (that is, all nodes of the document forest F in expressions $\iota(v, F)$) are fresh nodes that do not appear in any document where F will be inserted.

The semantics of a deterministic update on a document is clear. (In the case it speaks of a non-existing node, the update is simply ignored.) Formally:

Definition 7.2 Let $\tau = (o, 1)$ be a deterministic o-update operation and d a document. The *result* of the operation τ on d, denoted $\tau(d)$, is defined as follows:

1. d is unchanged if $v \notin \mathscr{V}(d)$ (for $o = \iota(v, F)$ or $o = \delta(v)$);
2. if $o = \iota(v, F)$ and $v \in \mathscr{V}(d)$, each tree in F is inserted as a child of v;
3. if $o = \delta(v)$ and $v \in \mathscr{V}(d)$, v is deleted (unless v is the root of d, in which case d is left unchanged).

More interestingly, we now define the semantics of an o-update operation on a px-space. Intuitively, a probabilistic o-update (o, c) performs the update on a document with probability c, and does nothing with probability $1 - c$.

Definition 7.3 Let $(\mathscr{D}, p)$ be a px-space and $\tau = (o, c)$ an o-update operation. The *result* of the operation τ on $(\mathscr{D}, p)$ is the px-space $(\mathscr{D}', p')$ where $\mathscr{D}' = \mathscr{D} \cup \{(o, 1)(d) \mid d \in \mathscr{D}\}$ and for each $d' \in \mathscr{D}'$:

$$p'(d') = p(d') \times (1 - c) + \sum_{\substack{d \in D \\ (o,1)(d) = d'}} (p(d) \times c).$$

We now consider the expressiveness of the different families of p-documents, which were presented in Section 4, with respect to probabilistic o-updates. The general question is, given a family of p-documents $\mathscr{F}$ and a probabilistic o-update operation τ, is the result of τ on the px-space associated with a p-document of $\mathscr{F}$ always representable as a p-document of $\mathscr{F}$? We define this next while taking tractability into consideration.

Definition 7.4 Let $\mathscr{F}$ be a family of p-documents. We say that $\mathscr{F}$ is *closed under* (respectively, *deterministic*) *o-updates* if for each (respectively, deterministic) o-update τ and for each $\tilde{\mathscr{P}} \in \mathscr{F}$, there exists a $\tilde{\mathscr{P}}' \in \mathscr{F}$ such that $\tau([\![\tilde{\mathscr{P}}]\!]) = [\![\tilde{\mathscr{P}}']\!]$. We say that $\mathscr{F}$ is *tractably* closed under o-updates if there exists a polynomial-time algorithm that, given a p-document $\tilde{\mathscr{P}} \in \mathscr{F}$ and an o-update operation τ, returns a p-document $\tilde{\mathscr{P}}' \in \mathscr{F}$ such that $[\![\tilde{\mathscr{P}}']\!] = \tau([\![\tilde{\mathscr{P}}]\!])$.

We can now study the closure of concrete families of p-document under o-updates, and the tractability of o-updates:

Proposition 7.5

1. *Every family of the form* $\mathsf{PrXML}^{\{\text{type}_1,\text{type}_2,\ldots\}}$ *or* $\mathsf{PrXML}^{\{\text{type}_1,\text{type}_2,\ldots\}}_{|\not\lambda}$ *(with* type_i *any of the types of distributional nodes defined in Section 4) is tractably closed under* ***deterministic*** *o-updates.*
2. *Every family of the form* $\mathsf{PrXML}^{\{\mathsf{ind},\text{type}_2,\text{type}_3,\ldots\}}$ *is tractably closed under o-updates.*
3. $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\lambda}$ *and* $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}_{|\not\lambda}$ *are tractably closed under o-updates.*
4. *A family* $\mathscr{F}$ *of p-documents such that* $\mathsf{PrXML}^{\{\}} \sqsubseteq_o \mathscr{F} \sqsubseteq_o \mathsf{PrXML}^{\{\mathsf{mux}\}}$ *is not closed under o-updates.*

Proof

1. Let $\mathscr{F}$ be such a family and $\tilde{\mathscr{P}} \in \mathscr{F}$. Let $(o, 1)$ be a deterministic o-update. Let $\tilde{\mathscr{P}}'$ be the result of applying o directly to the tree $\tilde{\mathscr{P}}$, as follows:
   - if $o = \iota(v, F)$ or $o = \delta(v)$, and $v \notin \mathscr{V}(\tilde{\mathscr{P}})$, then $\tilde{\mathscr{P}}' = \tilde{\mathscr{P}}$;
   - if $o = \iota(v, F)$ with v an ordinary node of $\tilde{\mathscr{P}}$, then we insert F as children of v in $\tilde{\mathscr{P}}$;
   - if $o = \delta(v)$ with v a non-root node of $\tilde{\mathscr{P}}$, we delete v from $\tilde{\mathscr{P}}$; additionally, if v is a child of a distributional node u, we adjust the probability of choosing its siblings:
     - if u is a det, ind, mux, or cie node, we do not change anything (in the case of mux, it means that the probability of not choosing any of the children of u increases by the amount $p^u(v)$);
     - if u is an exp node, we set $p'^u(W) = p^u(W) + p^u(W \cup \{v\})$.

   Then it is easy to see that $\tilde{\mathscr{P}}' \in \mathscr{F}$ and $[\![\tilde{\mathscr{P}}']\!] = (o, 1)([\![\tilde{\mathscr{P}}]\!])$.
2. Let $\mathscr{F}$ be such a family and $\tilde{\mathscr{P}} \in \mathscr{F}$. Let (o, c) be an o-update. We build $\tilde{\mathscr{P}}'$ from $\tilde{\mathscr{P}}$ as follows:
   - if $o = \iota(v, F)$ or $o = \delta(v)$, and $v \notin \mathscr{V}(\tilde{\mathscr{P}})$, then $\tilde{\mathscr{P}}' = \tilde{\mathscr{P}}$;
   - if $o = \iota(v, F)$ with v an ordinary node of $\tilde{\mathscr{P}}$, then we add as child of v an ind node v_1 that has for child another ind node v_2 that has for children the forest F, and we set $p'^{v_1}(v_2) = c$ and $p'^{v_2}(w_k) = 1$ for all w_k roots of F;
   - if $o = \delta(v)$ with v a non-root node of $\tilde{\mathscr{P}}$, we insert between v and its parent an ind note v' such that $p'^{v'}(v) = 1 - c$. If the parent of v is distributional, then the specifications of the probabilities are modified by replacing v with v'.

   Then $\tilde{\mathscr{P}}'$ is obviously an element of $\mathscr{F}$ and we can check that $[\![\tilde{\mathscr{P}}']\!] = \tau([\![\tilde{\mathscr{P}}]\!])$.
3. Let $\tilde{\mathscr{P}}$ be a document of either family. We can apply the update τ as in Part 2 above, yielding a document $\tilde{\mathscr{P}}' \in \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{exp},\mathsf{cie}\}}$ in which ind nodes have either a single child or are in effect det nodes. Either way, they can be transformed in polynomial time into exp nodes with probabilities specified for at most two subsets of children (the empty set and the full set), or into cie nodes. As already seen in the proof of Theorem 5.13, a hierarchy of cie nodes can be merged into a single cie node in polynomial time. There only remains the case of a hierarchy of exp nodes, with a succession of at most three exp nodes, two of which with probabilities specified for at most two subsets of children. This can be merged into a single exp node, with probabilities specified for at most $4k$ subsets, where k is the number of specifications of the third exp node.
4. Let $\tilde{\mathscr{P}}$ be a trivial p-document consisting of only one node u. This is a p-document of $\mathsf{PrXML}^{\{\}}$ and thus of $\mathscr{F}$. Let $\tau = (\iota(u, F), 0.5)$ be an o-update, where F is a forest that comprises only two nodes w and w' and no edges. Then $\tau([\![\tilde{\mathscr{P}}]\!]) = (\{d_1, d_2\}, p)$ where d_1 is a single-node tree and d_2 is a three-node tree, with $p(d_1) = 0.5$ and $p(d_2) = 0.5$. There is no possible way to represent this px-space in $\mathsf{PrXML}^{\{\mathsf{mux}\}}$ since the absence of siblings w and w' is correlated. □

Observe that for every two families $\mathscr{F}_1$ and $\mathscr{F}_2$ of p-documents, if $\mathscr{F}_1 \equiv_o \mathscr{F}_2$, then $\mathscr{F}_1$ is closed under o-updates if and only if $\mathscr{F}_2$ is closed under o-updates. Similarly, for $\mathscr{F}_1$ and $\mathscr{F}_2$ two families of p-documents, if $\mathscr{F}_1 \equiv_o^{\text{poly}} \mathscr{F}_2$, then $\mathscr{F}_1$ is tractably closed under o-updates if and only if $\mathscr{F}_2$ is tractably closed under o-updates. Note also that an immediate consequence of Proposition 7.5 and of the o-translations between families of p-documents obtained in Section 5 is the following characterization of the families closed under o-updates:

Corollary 7.6 *The following families are all tractably closed under o-updates:*

$\mathsf{PrXML}^{\{\mathsf{ind}\}}$, $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$, $\mathsf{PrXML}^{\{\mathsf{cie}\}}$, $\mathsf{PrXML}^{\{\mathsf{exp}\}}$, $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}$, $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\lambda}$ *and* $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}_{|\not\lambda}$.

However, $\mathsf{PrXML}^{\{\mathsf{mux}\}}$ *and* $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\lambda}$ *are not closed under o-updates.*

In other words, all reasonable probabilistic XML models are tractably closed under o-updates. However, value-based updates may in practice often be viewed as more "natural". We consider them next.

7.2 Value-Based Updates

Suppose we want to annotate all addresses in a document. We use a query to recognize addresses. We then

update all the nodes returned by the query. In general, a value-based update uses a *locator query* to locate the objects to update and then applies the update to all the objects. For example, the locators that are embedded in the XML update languages XUpdate [17] and XQuery Update [18] rely on XPath [19] and XQuery [20] queries, respectively. We consider simple query languages described later. First, we define queries abstractly.

Definition 7.7 A *locator query* Q is defined as a function that maps each document d to a set of pairs (d', v') where d' is an r-subtree of d and v' is a *locator*, that is, a node in d'. We assume that applying the same query to isomorphic documents yields isomorphic answers.

We can now define v-updates in terms of queries.

Definition 7.8 A probabilistic *v-update operation* is a pair $\tau = (o, c)$ where $0 < c \leq 1$ is the *confidence* in the operation, and o is either:

1. a *v-insertion*, that is, an expression $\iota(Q, F)$ where Q is a query and F is a document forest (to be inserted as children of the nodes mapped by Q);
2. a *v-deletion*, that is, an expression $\delta(Q)$ where Q is a query (indicating the nodes to delete).

If $c = 1$, τ is said to be *deterministic*.

Observe that again our updates are probabilistic. The locator query may introduce uncertainty, e.g., the system may make errors in recognizing addresses.

As in the discussion of Section 7.1 about o-updates, we consider the following two issues with respect to various families of p-documents: closure under v-updates, and the complexity of computing a representation of the result of an update. We first define the semantics of deterministic v-updates on ordinary documents.

Definition 7.9 Let $\tau = (o, 1)$ be a (deterministic) v-update operation and d a document. The *result* of the operation τ on d, denoted $\tau(d)$, is the result of applying o to d:

1. if $o = \iota(Q, F)$, each tree in F is inserted as a child of each v' such that some $(d', v') \in Q(d)$ (possibly inserting F multiple times at the same place);
2. if $o = \delta(Q)$, each v' such that some $(d', v') \in Q(d)$ is deleted.

The definition of the result of a v-update operation on a px-space is a straightforward adaptation of Definition 7.3 for o-updates. Closure under v-updates, however, uses px-space isomorphism instead of equality:

Definition 7.10 Let $\mathscr{F}$ be a family of p-documents. We say that $\mathscr{F}$ is *closed under v-updates* (respectively, *deterministic v-updates*) for the class of queries $\mathscr{Q}$ if, for any v-update (respectively, deterministic v-update) $\tau = (o, c)$ with o defined by a query $Q \in \mathscr{Q}$, for each $\tilde{\mathscr{P}} \in \mathscr{F}$, there exists a $\tilde{\mathscr{P}}' \in \mathscr{F}$ such that $\tau([\![\tilde{\mathscr{P}}]\!]) \sim [\![\tilde{\mathscr{P}}']\!]$. We say $\mathscr{F}$ is *tractably closed under v-updates* for the class $\mathscr{Q}$ if there is an algorithm that returns such a $\tilde{\mathscr{P}}'$ given $\tilde{\mathscr{P}}$ in time *polynomial in the size of* $\tilde{\mathscr{P}}$.[8]

We next introduce three classes of queries that we will consider for closure and tractability results. The first one is the class of tree-pattern queries, e.g., queries of the form `a[b/c][d]`. This is one of the most studied classes of queries for XML. We use here for simplicity a restricted notion of tree-pattern queries, without descendant edges (the `//` of XPath). This class can be extended in a straightforward manner. But, as we shall see, even simple branching as considered here leads to negative results. We also consider a simpler class, namely that of restricted single-path queries, e.g., queries of the form `/a/b/c`. Finally, we consider a more abstract class, namely the "locally monotone queries" that includes the tree-pattern queries. We will show for that class a very strong positive result (the tractable closure of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ under v-insertions defined by locally monotone queries).

Definition 7.11

1. A *tree-pattern query* Q is defined by an underlying tree-pattern d_Q (which is simply an ordinary document) and a locator node $v_Q \in \mathscr{V}(d_Q)$. For a document d, $Q(d)$ is the set of all pairs $\{d', v'\}$ such that d' is an r-subtree of d, there is a homomorphism from d_Q to d', and v' is the image of v_Q by this homomorphism.
2. A *single-path query* is a tree-pattern query such that the underlying tree-pattern is a single path without branching. A *restricted single-path query* Q is a single-path query whose locator node is the terminal node v_Q of the path.
3. A query Q is *locally monotone* if either of the following two equivalent conditions holds:
   (i) For any three documents d_1, d_2 and d_3 such that d_1 is an r-subtree of d_2 and d_2 is a r-subtree of d_3, $(d_1, v) \in Q(d_2) \iff (d_1, v) \in Q(d_3)$;
   (ii) For any two documents d_1 and d_2 such that d_1 is an r-subtree of d_2, $Q(d_1)$ is the subset of elements of $Q(d_2)$ that are r-subtrees of d_1.

The previous definition is well-defined because 3i and 3ii are equivalent as we briefly argue next:

[8] We consider only here the *data complexity*, i.e., the query is not considered to be part of the input.

3i ⇒ 3ii. Let d_1 and d_2 be two documents such that d_1 is an r-subtree of d_2. Let $(d', v') \in Q(d_1)$. By definition of queries, d' is an r-subtree of d_1. By 3i, $(d', v') \in Q(d_2)$. Now let $(d', v') \in Q(d_2)$ with d' an r-subtree of d_1. By 3i, $(d', v') \in Q(d_1)$. This concludes the proof of the implication.

3ii ⇒ 3i. Let d_1, d_2 and d_3 be three documents such that d_1 is an r-subtree of d_2 and d_2 an r-subtree of d_3. Suppose first that $(d_1, v') \in Q(d_3)$. As d_1 is an r-subtree of d_2, by 3ii, $(d_1, v') \in Q(d_2)$. Now suppose $(d_1, v') \in Q(d_2)$. By 3ii, $(d_1, v') \in Q(d_3)$.

Locally monotone queries actually generalize tree-pattern queries:

Proposition 7.12 *Every tree-pattern query is locally monotone.*

Proof We prove (ii) of Definition 7.11. Let Q be a tree-pattern query defined by the pattern d_Q and locator v_Q. Let d_1 and d_2 be two documents, such that d_1 is an r-subtree of d_2. Let $(d', v') \in Q(d_1)$. By definition, d' is an r-subtree of d_1. As there is a homomorphism from d_Q to d' mapping v_Q to v' and d' is an r-subtree of d_2, $(d', v') \in Q(d_2)$. Now let $(d', v') \in Q(d_2)$ be an r-subtree of d_1. Then there is a homomorphism from d_Q to d' mapping v_Q to v', so $(d', v') \in Q(d_1)$. □

We showed in [8] that tree-pattern queries with descendant edges and value joins (both positive and negative) are still locally monotone. On the other hand, a simple query such as "Return the root if all its children are labeled by l" is not locally monotone because the universal quantifier involves some form of negation. We present now basic results about closure under v-updates and tractability.

Proposition 7.13

1. *Let $\mathscr{F}$ be a family of p-documents that is closed with respect to relabeling of ordinary nodes. If $\mathscr{F}$ is closed under v-updates (respectively, deterministic v-updates) for the class of restricted single-path queries, then it is closed under o-updates (respectively, deterministic o-updates).*
2. *If $\mathscr{F}$ is a family of documents (tractably) closed under* deterministic *o-updates, then it is (tractably) closed under deterministic v-updates for restricted single-path queries.*
3. *Let $\mathscr{F}_1$ and $\mathscr{F}_2$ be two families of p-documents such that $\mathscr{F}_1 \equiv_v \mathscr{F}_2$. Then, $\mathscr{F}_1$ is closed under v-updates if and only if $\mathscr{F}_2$ is closed under v-updates.*
4. *Let $\mathscr{F}_1$ and $\mathscr{F}_2$ be two families of p-documents closed under v-updates for some class of queries $\mathscr{Q}$, such that $\mathscr{F}_1 \equiv_v^{\text{poly}} \mathscr{F}_2$. Then $\mathscr{F}_1$ is tractably closed under v-updates for the class of queries $\mathscr{Q}$ if and only if $\mathscr{F}_2$ is tractably closed under v-updates for the class of queries $\mathscr{Q}$.*
5. *Any family $\mathscr{F}$ that satisfies $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}} \sqsubseteq_v \mathscr{F}$ is closed under v-updates for any class of queries.*

Proof

1. Suppose $\mathscr{F}$ is closed under v-updates (the proof is the same for deterministic v-updates) for the class of restricted single-path queries. Let $\tilde{\mathscr{P}} \in \mathscr{F}$ and τ_o be an o-update. Let φ be a function that maps each ordinary node of $\tilde{\mathscr{P}}$ to a unique label (φ^{-1} then maps these identifiers to nodes of $\tilde{\mathscr{P}}$). τ_o is defined by a node v. If this node is not in $\tilde{\mathscr{P}}$, then $\tau_o([\![\tilde{\mathscr{P}}]\!]) = [\![\tilde{\mathscr{P}}]\!]$. Otherwise, let $\tilde{\mathscr{P}}_v$ be the relabeling of $\tilde{\mathscr{P}}$ by φ and d the minimal r-subtree of $\mathsf{doc}(\tilde{\mathscr{P}}')$ containing the node labeled by $\varphi(v)$ (this is obviously a single path). Let τ_v be the v-update corresponding to τ_o where the locator v is replaced by the restricted single-path query d with locator the node labeled by $\varphi(v)$. Since $\mathscr{F}$ is closed under v-updates, there is a document $\tilde{\mathscr{P}}'_v$ such that $[\![\tilde{\mathscr{P}}'_v]\!] = \tau_v([\![\tilde{\mathscr{P}}_v]\!])$. Observe now that if one apply φ^{-1} to the labels of the nodes of $\tau_v([\![\tilde{\mathscr{P}}_v]\!])$ to get back original nodes of $\tilde{\mathscr{P}}$, one obtains exactly $\tau_o([\![\tilde{\mathscr{P}}_v]\!])$ since τ_o and τ_v both perform the same update at the same place. Let now $\tilde{\mathscr{P}}'$ be the p-document obtained from $\tilde{\mathscr{P}}'_v$ by applying φ^{-1} to the labels to get back original nodes of $\tilde{\mathscr{P}}$. Then $[\![\tilde{\mathscr{P}}']\!] = \tau_o([\![\tilde{\mathscr{P}}]\!])$. Note that this construction might not be polynomial even if $\mathscr{F}$ is tractably closed under v-updates for the class of restricted single-path queries, since the query defining τ_v is not fixed.
2. Suppose $\mathscr{F}$ is closed under deterministic o-updates. Let $\tilde{\mathscr{P}} \in \mathscr{F}$ and $\tau_v = (o_v, c_v)$ be a deterministic v-update defined by restricted single-path query Q. Let S be the set of answers of Q on $\mathsf{doc}(\tilde{\mathscr{P}})$. Since Q is a single-path query the number of elements in $|S|$ is at most the number of ordinary nodes in $\tilde{\mathscr{P}}$. For each $\{(d', v')\} \in S$, we define the deterministic o-update $\tau_o^{(d',v')}$ that performs the same update operation as τ_v except that the locator query is replaced by v'. We apply now the o-updates $\tau_o^{(d',v')}$ for each $\{(d', v')\}$ sequentially on $[\![\tilde{\mathscr{P}}]\!]$, yielding a px-space $(\mathscr{D}', p')$. Observe that the ordering of these o-updates is not significant and that $(\mathscr{D}', p')$ is exactly $\tau_v([\![\tilde{\mathscr{P}}]\!])$. As $\mathscr{F}$ is closed under deterministic o-updates, there is a $\tilde{\mathscr{P}}' \in \mathscr{F}$ such that $[\![\tilde{\mathscr{P}}']\!] = (\mathscr{D}', p')$. The construction of $\tilde{\mathscr{P}}'$ from $\tilde{\mathscr{P}}$ is polynomial if $\mathscr{F}$ is tractably closed under o-updates (we use here the bound on $|S|$).
3. Suppose $\mathscr{F}_1$ is closed under v-updates. Let $\tilde{\mathscr{P}}_2 \in \mathscr{F}_2$ and τ be a v-update. Since $\mathscr{F}_2 \sqsubseteq_v \mathscr{F}_1$, there exists $\tilde{\mathscr{P}}_1 \in \mathscr{F}_1$ such that $[\![\tilde{\mathscr{P}}_1]\!] \sim [\![\tilde{\mathscr{P}}_2]\!]$. As $\mathscr{F}_1$ is

closed under v-updates, there exists $\tilde{\mathscr{P}}'_1 \in \mathscr{F}_1$ such that $[\![\tilde{\mathscr{P}}'_1]\!] \sim \tau([\![\tilde{\mathscr{P}}_1]\!])$. Since $\mathscr{F}_1 \sqsubseteq_v \mathscr{F}_2$, there exists $\tilde{\mathscr{P}}'_2 \in \mathscr{F}_2$ such that $[\![\tilde{\mathscr{P}}'_2]\!] \sim [\![\tilde{\mathscr{P}}'_1]\!]$. Then $[\![\tilde{\mathscr{P}}'_2]\!] \sim \tau([\![\tilde{\mathscr{P}}_2]\!])$, and $\mathscr{F}_2$ is closed under v-updates. The other direction is obtained by symmetry.
4. This is proved as in 3, since $\tilde{\mathscr{P}}_1$, $\tilde{\mathscr{P}}'_1$ and $\tilde{\mathscr{P}}'_2$ can be obtained in polynomial time from, respectively, $\tilde{\mathscr{P}}_2$, $\tilde{\mathscr{P}}_1$ and $\tilde{\mathscr{P}}'_1$ (observe that the query defining τ remains fixed).
5. This is a direct consequence of Proposition 5.7. □

We now consider the tractability of v-updates for families such as $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ and $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. The following result shows that, at least for v-insertions, the ability of expressing complex dependencies through cie nodes makes a difference in the complexity of updates.

Theorem 7.14

1. *Every family of the form* $\mathsf{PrXML}^{\{\mathrm{type}_1,\mathrm{type}_2,\ldots\}}$ *or* $\mathsf{PrXML}^{\{\mathrm{type}_1,\mathrm{type}_2,\ldots\}}_{|\not\downarrow}$ *with* type_i *any of the types of distributional nodes defined in Section 4, except* cie*, is not tractably closed under deterministic v-insertions defined by single-path queries.*
2. $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ *is tractably closed under v-insertions defined by locally monotone queries, as long as computing query results can be done in polynomial time.*[9]

Proof

1. With the exception of such families as $\mathsf{PrXML}^{\{\}}$ that are not even closed under o-updates, $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$ is efficiently v-translatable to each considered family $\mathscr{F}$. For one such $\mathscr{F}$, and for an arbitrary positive integer n, let $\tilde{\mathscr{P}}$ be the efficient v-translation in $\mathscr{F}$ of the p-document of $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$ shown in Figure 7.1(a). We only show here node labels, not node identifiers. Let τ be the deterministic v-insertion defined by the single-path pattern of Figure 7.1(b), that inserts a single node labeled by c as a child of the root node a.
   Suppose that $\tilde{\mathscr{P}}'$ is a p-document of $\mathscr{F}$ such that $[\![\tilde{\mathscr{P}}']\!] \sim \tau([\![\tilde{\mathscr{P}}]\!])$. We proceed very similarly to the proof of Theorem 5.15: as the number of existing c nodes needs to be correlated with the number of existing b_i nodes, all distributional nodes appearing below a b child must yield possible worlds with a fixed number of b' children and must thus appear above all nodes labeled by b'. This means that, for a given value of k (say, $k = n/2$, assuming n is even), each possible choice of k b_i nodes among n must appear as an ordinary subtree in $\tilde{\mathscr{P}}$. But

$$\binom{n}{n/2} = \frac{n!}{(n/2)!^2} \sim \frac{\sqrt{2\pi n} n^n e^n}{e^n \pi n \times (n/2)^n} = \frac{2^n\sqrt{2}}{\sqrt{\pi n}} = \\ = \Omega(2^n)$$

   using Stirling's formula.
2. This has been proved in [8]. As the proof requires a number of intermediate results (especially on the possibility of applying locally monotone queries directly to p-documents of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$), we only describe here the general idea. Given a v-insertion τ defined by a query Q and a p-document $\tilde{\mathscr{P}}$, we apply Q directly to $\mathsf{doc}(\tilde{\mathscr{P}})$, keeping for each query result r the set of event conjunctions $cond_r$ on nodes appearing in the query result. Then, for each query result r, the nodes to be inserted are inserted at the place indicated by the locator, under a fresh cie node, with the conjunction of $cond_r$ as the condition. Because Q is locally monotone, it can be shown that this process yields a p-document $\tilde{\mathscr{P}}'$ such that $[\![\tilde{\mathscr{P}}']\!] \sim \tau([\![\tilde{\mathscr{P}}]\!])$. Besides, it is obviously a polynomial-time process, as long as Q takes polynomial time on $\mathsf{doc}(\tilde{\mathscr{P}})$. □

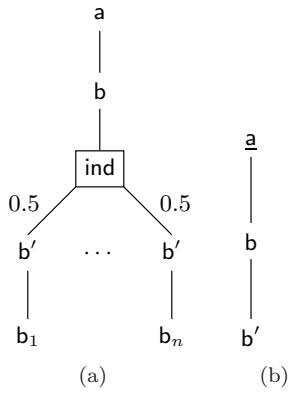

Fig. 7.1 A p-document (a) of $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\downarrow}$ on which a v-insertion defined by single-path pattern (b) can result in exponential blowup

It is an open issue whether $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ is tractably closed under arbitrary v-updates (including deletions) defined by tree-pattern queries. We have shown in [8], however, that v-updates are intractable in $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ if we impose the result of an update to be expressed with the same events as in the original document (this is usually what we want when updating $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ p-documents, since this allows the keeping of *lineage* or *provenance* information, each event being a trace of the update that introduced it).

[9] This is especially the case for tree-pattern queries, whose *data* complexity is polynomial-time.

8 Conclusion

Under the object-based semantics, $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}$ is the most expressive family (among those studied) and has two crucial properties. It is tractably closed under o-updates, and all the other families can be efficiently o-translated into it (but the converse is not true). Under the value-based semantics, $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}$ remains the most expressive. Notwithstanding, other families, including $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$, $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ and $\mathsf{PrXML}^{\{\mathsf{cie}\}}$, are as expressive as $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}$. V-translations from $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ into either $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ or $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ may entail an exponential blowup in the size of the p-document. It is unknown whether there are efficient v-translations from $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ into $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ and $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. Nonetheless, p-documents of $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ with a bounded distributional depth or out-degree can be efficiently v-translated into the other two families. As for updates, v-insertions (defined by locally monotone queries) are tractable for $\mathsf{PrXML}^{\{\mathsf{cie}\}}$, but not for $\mathsf{PrXML}^{\{\mathsf{exp}\}}$. Therefore, under the value-based semantics, $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ has the advantage in terms of insertions, the ability to efficiently translate into it, and the power to express correlations between different distributional nodes. However, tree-pattern queries with projection can be evaluated efficiently (under data complexity) in the family $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ [9], and even in $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ [1, 10], but (except for trivial cases) they are #P-hard in $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ [1, 10]. Thus, the choice of a probabilistic XML model hinges on a trade-off between efficient query processing and the ability to capture complex correlations.

We conclude by discussing some extensions. In [21], the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ is enriched with constraints that make it possible to express correlations between distributional nodes, without sacrificing the efficiency of query evaluation; however, update tractability is still open. In [22], p-documents are extended by allowing order among siblings. Alternatively, one might consider two p-documents to be the same if there are homomorphisms in both directions; the effect on translatability and updates is left for future work. Finally, it is important to study the complexity of some additional problems, such as testing equivalence of p-documents and enumerating all random documents that have probability above a given threshold. In particular, it would be interesting to find out how these complexities depend on the types of distributional nodes being used.

References

1. Kimelfeld, B., Kosharovski, Y., Sagiv, Y.: Query efficiency in probabilistic XML models. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM (2008)
2. Senellart, P., Abiteboul, S.: On the complexity of managing probabilistic XML data. In: Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 283–292. ACM (2007)
3. Nierman, A., Jagadish, H.V.: ProTDB: Probabilistic data in XML. In: VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases, pp. 646–657. Morgan Kaufmann (2002)
4. Hung, E., Getoor, L., Subrahmanian, V.S.: PXML: A probabilistic semistructured data model and algebra. In: Proceedings of the 19th International Conference on Data Engineering, pp. 467–478 (2003)
5. Hung, E., Getoor, L., Subrahmanian, V.S.: Probabilistic interval XML. ACM Transactions on Computational Logic **8**(4) (2007)
6. van Keulen, M., de Keijzer, A., Alink, W.: A probabilistic XML approach to data integration. In: Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, pp. 459–470. IEEE Computer Society (2005)
7. Abiteboul, S., Senellart, P.: Querying and updating probabilistic information in XML. In: Advances in Database Technology - EDBT 2006, 10th International Conference on Extending Database Technology, *Lecture Notes in Computer Science*, vol. 3896, pp. 1059–1068. Springer (2006)
8. Senellart, P.: Comprendre le Web caché. Understanding the Hidden Web. Ph.D. thesis, Université Paris-Sud 11 (2007)
9. Kimelfeld, B., Sagiv, Y.: Matching twigs in probabilistic XML. In: Proceedings of the Thirty Third International Conference on Very Large Data Bases (VLDB), pp. 27–38. ACM (2007)
10. Kimelfeld, B., Kosharovsky, Y., Sagiv, Y.: Query evaluation over probabilistic XML. The VLDB Journal (2009)
11. Li, T., Shao, Q., Chen, Y.: PEPX: a query-friendly probabilistic XML database. In: CIKM, pp. 848–849. ACM (2006)
12. Dalvi, N.N., Suciu, D.: Management of probabilistic data: foundations and challenges. In: PODS, pp. 1–12 (2007)
13. Widom, J.: Trio: A system for integrated management of data, accuracy, and lineage. In: CIDR, pp. 262–276 (2005)
14. Koch, C.: MayBMS: A system for managing large uncertain and probabilistic databases. In: C. Aggarwal (ed.) Managing and Mining Uncertain Data. Springer-Verlag (2009)
15. Imielinski, T., Jr., W.L.: Incomplete information in relational databases. Journal of the ACM **31**(4), 761–791 (1984)
16. Green, T.J., Tannen, V.: Models for incomplete and probabilistic information. In: Current Trends in Database Technology - EDBT 2006, EDBT 2006 Workshops PhD, DataX, IIDB, IIHA, ICSNW, QLQP, PIM, PaRMA, and Reactivity on the Web, *Lecture Notes in Computer Science*, vol. 4254, pp. 278–296. Springer (2006)
17. XML::DB Initiative: XUpdate. `http://xmldb-org.sourceforge.net/xupdate/` (2000). Working Draft
18. W3C: XQuery Update facility. `http://www.w3.org/TR/xquery-update-10/` (2008). Candidate Recommandation
19. W3C: XML Path language (XPath). `http://www.w3.org/TR/xpath` (1999). Recommandation
20. W3C: XQuery 1.0: An XML query language. `http://www.w3.org/TR/xquery/` (2007). Recommandation
21. Cohen, S., Kimelfeld, B., Sagiv, Y.: Incorporating constraints in probabilistic XML. In: Proceedings of the Twenty-Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS) (2008)
22. Cohen, S., Kimelfeld, B., Sagiv, Y.: Running tree automata on probabilistic XML (2009). To appear in *Proceedings of the Twenty-Eigthth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*