# Efficient Global Probabilistic Deduction from Taxonomic and Probabilistic Knowledge-Bases over Conjunctive Events

Thomas Lukasiewicz

Institut für Informatik, Universität Gießen
Arndtstr. 2, D-35392 Gießen, Germany
Tel: +49-641-99-32153, Fax: +49-641-99-32149
E-mail: lukasiewicz@informatik.uni-giessen.de

## Abstract

We present a new, efficient linear programming approach to probabilistic deduction from probabilistic knowledge-bases over conjunctive events. We show that this approach enables us to solve the classical problem of probabilistic deduction along a chain of basic events in polynomial time in the length of the chain. We then elaborate how taxonomic knowledge can be exploited in our new approach for an increased efficiency. We also present important new results for the classical linear programming approach to probabilistic deduction under taxonomic knowledge.

## 1 Introduction

There are many approaches to non-Bayesian probabilistic deduction in the literature. They can be classified in global techniques based on linear programming and in local methods founded on the iterative application of inference rules.

Non-Bayesian probabilistic deduction by solving linear programs is discussed e.g. in [23], [13], [24], [17], [14], [2], [15], and [22]. It can be performed within rich probabilistic languages capable of representing many facets of probabilistic knowledge (see especially [14]). Crucially, it guarantees the global precision of probabilistic deductions. However, all global techniques developed so far run in exponential time in the size of the probabilistic deduction problems.

Mainly this lack of efficiency of global techniques encouraged many researchers to work on local methods (see e.g. [8], [9], [1], [11], [28], [16], and [19]). Probabilistic deductions by local techniques are generally performed from more restricted languages. These restrictions are necessary if attempting to gain global precision and efficiency. The local approach generally does not guarantee the global precision of probabilistic deductions. However, it is commonly expected to produce results of sufficient precision. Furthermore, it is expected to be more efficient than global techniques.

Even though local techniques have been considered very promising in the literature so far, we showed in [20] that they are extremely limited in their application. In detail, we analyzed the precision and the efficiency of local methods in the framework of taxonomic and probabilistic knowledge-bases over conjunctive events. As a surprising result, we discovered that the global precision of probabilistic deductions cannot be guaranteed at all. We gave examples of extremely imprecise probabilistic deductions. Furthermore, we showed that the presented local approach is not more efficient than the classical linear programming approach for that framework.

This paper deals again with taxonomic and probabilistic knowledge-bases over conjunctive events. However, it analyzes the deduction problem from the viewpoint of linear programming. We present a new, efficient linear programming approach to probabilistic deduction from probabilistic knowledge-bases over conjunctive events. We show that it solves the classical problem of probabilistic deduction along a chain of basic events (see especially [9], [1], [28], and [20]) in a very efficient and elegant way.

As in [20], we are also concerned about probabilistic deduction under taxonomic knowledge in this paper. We elaborate how taxonomic knowledge can be exploited for an increased efficiency in both our new and the classical approach to probabilistic deduction by linear programming. The interplay between taxonomic and probabilistic knowledge is also analyzed in [16], where probabilistic knowledge is integrated into a terminological language.

The rest of this paper is organized as follows. Section 2 deals with the formal background. Section 3 presents our new linear programming approach to probabilistic deduction. Section 4 generalizes to probabilistic deduction under taxonomic knowledge. In Section 5, we provide efficient algorithms for computing the variables of the generated linear programs. Section 6 discusses possible extensions. Section 7 sums up the main results and gives an outlook on possible directions of future research.

## 2 Taxonomic and Probabilistic Knowledge

We focus on taxonomic and probabilistic formulas over conjunctive events. The set of conjunctive events comprises the *false event* $\perp$, the *true event* $\top$, all *basic events* taken from a finite and nonempty set $\mathcal{B}$ and all conjunctions of more than two basic events. Taxonomic formulas describe implications between conjunctive events. Probabilistic formulas represent interval restrictions for conditional probabilities of conjunctive events.

For technical convenience, we choose a set-oriented definition of *conjunctive events* as members of $\mathcal{C}_{\mathcal{B}} = 2^{\mathcal{B}} \cup \{\mathcal{B} \cup \{\bot\}\}$. The classical notation is used as an abbreviation: $\bot$, $\top$, and $B_1 B_2 \ldots B_k$ with $k \geq 1$ abbreviate the conjunctive events $\mathcal{B} \cup \{\bot\}$, $\emptyset$, and $\{B_1, B_2, \ldots, B_k\}$, respectively. Furthermore, the expression $C_1 C_2 \ldots C_k$ with $k \geq 2$ abbreviates the union $C_1 \cup C_2 \cup \cdots \cup C_k$ of the conjunctive events $C_1, C_2, \ldots, C_k$. *Taxonomic* and *probabilistic formulas* are expressions of the form $C \to D$ and $(D|C)[u_1, u_2]$, respectively, with $C, D \in \mathcal{C}_{\mathcal{B}}$ and $0 \leq u_1 \leq u_2 \leq 1$.

To define probabilistic interpretations of conjunctive events, taxonomic formulas, and probabilistic formulas, we introduce atomic events and the binary relation $\preceq$ between atomic and conjunctive events. The atomic events of our framework coincide with the more commonly known possible worlds from Nilsson's probabilistic logic [23]. Again for technical convenience, we define *atomic events* as members of $\mathcal{A}_{\mathcal{B}} = 2^{\mathcal{B}}$ and use their classical notation as an abbreviation: $E_1 E_2 \ldots E_n$ with $\{E_1, E_2, \ldots, E_n\} = \{B_1, B_2, \ldots, B_k\} \dot{\cup} \{\overline{B}_{k+1}, \overline{B}_{k+2}, \ldots, \overline{B}_n\}$ and $\{B_1, B_2, \ldots, B_n\} = \mathcal{B}$ abbreviates the atomic event $\{B_1, B_2, \ldots, B_k\}$. For all atomic events $A$ and all conjunctive events $C$, let $A \preceq C$ iff $C \subseteq A$.

A *probabilistic interpretation* is a mapping $\mathrm{P}\colon \mathcal{A}_{\mathcal{B}} \longrightarrow [0,1]$ such that all $\mathrm{P}(A)$ with $A \in \mathcal{A}_{\mathcal{B}}$ sum up to 1. It is extended to conjunctive events $C$ by $\mathrm{P}(C) = \sum_{A \in \mathcal{A}_{\mathcal{B}},\, A \preceq C} \mathrm{P}(A)$. It is extended to taxonomic and probabilistic formulas by:

$$\begin{array}{lll} \mathrm{P} \models C \to D & \text{iff} & \mathrm{P}(C) = \mathrm{P}(CD)\,, \\ \mathrm{P} \models (D|C)[u_1, u_2] & \text{iff} & u_1 \cdot \mathrm{P}(C) \leq \mathrm{P}(CD) \leq u_2 \cdot \mathrm{P}(C)\,. \end{array}$$

A probabilistic interpretation P is a *model* of a formula $F$ iff $\mathrm{P} \models F$. P is a *model* of a set of formulas $\mathcal{F}$, denoted $\mathrm{P} \models \mathcal{F}$, iff P is a model of all $F \in \mathcal{F}$. A set of formulas $\mathcal{F}$ is *satisfiable* iff a model of $\mathcal{F}$ exists. A formula $F$ is a *logical consequence* of a set of formulas $\mathcal{F}$, denoted $\mathcal{F} \models F$, iff each model of $\mathcal{F}$ is also a model of $F$. Assuming

$$\boldsymbol{u} = \{u \in [0,1] \mid \exists\, \mathrm{P} : \mathrm{P} \models \mathcal{F} \cup \{(H|G)[u,u]\},\ \mathrm{P}(G) > 0\}$$

for a probabilistic formula $(H|G)[u_1, u_2]$ and a set of formulas $\mathcal{F}$, it can easily be verified that $(H|G)[u_1, u_2]$ is a logical consequence of $\mathcal{F}$ iff $\mathcal{F} \not\models G \to \bot$ entails

$$u_1 \ \leq\ \inf \boldsymbol{u} \quad \text{and} \quad u_2 \ \geq\ \sup \boldsymbol{u}\,.$$

This observation yields a canonic notion of precision for logical consequences of probabilistic formulas. The probabilistic formula $(H|G)[u_1, u_2]$ is a *precise logical consequence* of $\mathcal{F}$, denoted $\mathcal{F} \models_{prec} (H|G)[u_1, u_2]$, iff $\mathcal{F} \not\models G \to \bot$ entails

$$u_1 \ =\ \inf \boldsymbol{u} \quad \text{and} \quad u_2 \ =\ \sup \boldsymbol{u}\,.$$

A *taxonomic and probabilistic knowledge-base* $(\mathcal{B}, \mathcal{T}, \mathcal{P})$ consists of a set of basic events $\mathcal{B}$, a set of taxonomic formulas $\mathcal{T}$ over $\mathcal{C}_{\mathcal{B}}$, and a set of probabilistic formulas $\mathcal{P}$ over $\mathcal{C}_{\mathcal{B}}$. A taxonomic and probabilistic knowledge-base $(\mathcal{B}, \mathcal{T}, \mathcal{P})$ with $\mathcal{T} = \emptyset$ is abbreviated by $(\mathcal{B}, \mathcal{P})$ and shortly called *probabilistic knowledge-base*. A *probabilistic query* to a taxonomic and probabilistic knowledge-base $(\mathcal{B}, \mathcal{T}, \mathcal{P})$ is an expression $?(F|E)[x_1, x_2]$ with $E, F \in \mathcal{C}_{\mathcal{B}}$ and variables $x_1, x_2$ with $x_1 \neq x_2$. Its *precise answer* is $[x_1, x_2] = [u_1, u_2]$ if $\mathcal{T} \cup \mathcal{P} \not\models E \to \bot$ and $\mathcal{T} \cup \mathcal{P} \models_{prec} (F|E)[u_1, u_2]$, its precise answer is $E \to \bot$ if $\mathcal{T} \cup \mathcal{P} \models E \to \bot$.

We provide a small example to illustrate the introduced definitions. The taxonomic knowledge that all ostriches are birds and that ostriches do not fly, and the probabilistic knowledge that at least 90 per cent of all birds fly and that not more that 10 per cent of all birds are ostriches can be expressed by the following taxonomic and probabilistic knowledge-base:

$$(\{\mathsf{os}, \mathsf{bi}, \mathsf{fl}\}, \{\mathsf{os} \to \mathsf{bi},\ \mathsf{os\,fl} \to \bot\}, \{(\mathsf{fl}|\mathsf{bi})[.9, 1], (\mathsf{os}|\mathsf{bi})[0, .1]\})\,.$$

Wondering about the precise lower and upper bounds of the relative frequency of ostriches among the birds, we can formulate the probabilistic query $?(\mathsf{os}|\mathsf{bi})[x_1, x_2]$ yielding the precise answer $[x_1, x_2] = [0, .1]$.

After having introduced the necessary formal background, we can describe the main topic of this paper as the problem of computing the precise answer to a probabilistic query.

## 3 Efficient Probabilistic Deduction

In this section, we present a new, efficient approach to probabilistic deduction by linear programming. We start by describing the key ideas by some intuitive examples.

### 3.1 The Key Ideas

Let the probabilistic knowledge-base $(\mathcal{B}, \mathcal{P})$ be given by $\mathcal{B} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}\}$ and $\mathcal{P} = \{(\mathsf{B}|\mathsf{A})[0, .2], (\mathsf{C}|\mathsf{AB})[0, .2], (\mathsf{D}|\mathsf{C})[0, .2]\}$. Let us consider the probabilistic query $?(\mathsf{BCD}|\mathsf{A})[x_1, x_2]$.

Assuming that we already verified $\mathcal{P} \not\models \mathsf{A} \to \bot$, we could naively think that the questioned precise bounds can be computed by solving the following two linear fractional programs with $p_C \geq 0$ ($C \in \{\top, \mathsf{A}, \mathsf{AB}, \mathsf{ABC}, \mathsf{ABCD}, \mathsf{C}, \mathsf{CD}\}$) and $\mathrm{opt} \in \{\min, \max\}$:

$$\begin{array}{lll} \multicolumn{3}{l}{\mathrm{opt}\ \frac{p_{\mathsf{ABCD}}}{p_{\mathsf{A}}}\ \text{subject to}} \\ p_{\top} & = & 1 \\ p_{\mathsf{AB}} & \leq & 0.2 \cdot p_{\mathsf{A}} \\ p_{\mathsf{ABC}} & \leq & 0.2 \cdot p_{\mathsf{AB}} \\ p_{\mathsf{CD}} & \leq & 0.2 \cdot p_{\mathsf{C}} \\ p_{\mathsf{A}} & > & 0 \end{array} \tag{1}$$

Of course, the questioned precise bounds are in general not given by the optimal values of these two linear fractional programs. This is due to the fact that the considered linear constraints do not describe the implicit taxonomic relationships between the conjunctive events attached as index to the variables. Hence, for an optimal solution $p_C$ of (1), it does not necessarily exist a corresponding model P of $\mathcal{P}$ with $\mathrm{P}(C) = p_C$ for $C \in \{\top, \mathsf{A}, \mathsf{AB}, \mathsf{ABC}, \mathsf{ABCD}, \mathsf{C}, \mathsf{CD}\}$.

The classical way to describe these taxonomic relationships (see e.g. [23] and [24]) is to express each conjunctive event $C$ that appears as index in (1) by the disjunction of all atomic events that imply $C$. Thus, our first try to compute the questioned precise bounds is corrected by adding the following linear constraints with $v_A \geq 0$ ($A \in \mathcal{A}_{\mathcal{B}}$) to (1):

$$\begin{pmatrix} p_{\top} \\ p_{\mathsf{A}} \\ p_{\mathsf{AB}} \\ p_{\mathsf{ABC}} \\ p_{\mathsf{ABCD}} \\ p_{\mathsf{C}} \\ p_{\mathsf{CD}} \end{pmatrix} = \begin{pmatrix} 1111\ 1111\ 1111\ 1111 \\ 1111\ 1111\ 0000\ 0000 \\ 1111\ 0000\ 0000\ 0000 \\ 1100\ 0000\ 0000\ 0000 \\ 1000\ 0000\ 0000\ 0000 \\ 1100\ 1100\ 1100\ 1100 \\ 1000\ 1000\ 1000\ 1000 \end{pmatrix} \cdot \begin{array}{l} (v_{\mathsf{ABCD}}, v_{\mathsf{ABC\overline{D}}}, v_{\mathsf{AB\overline{C}D}}, v_{\mathsf{AB\overline{C}\,\overline{D}}}, \\ v_{\mathsf{A\overline{B}CD}}, v_{\mathsf{A\overline{B}C\overline{D}}}, v_{\mathsf{A\overline{B}\,\overline{C}D}}, v_{\mathsf{A\overline{B}\,\overline{C}\,\overline{D}}}, \\ v_{\mathsf{\overline{A}BCD}}, v_{\mathsf{\overline{A}BC\overline{D}}}, v_{\mathsf{\overline{A}B\overline{C}D}}, v_{\mathsf{\overline{A}B\overline{C}\,\overline{D}}}, \\ v_{\mathsf{\overline{A}\,\overline{B}CD}}, v_{\mathsf{\overline{A}\,\overline{B}C\overline{D}}}, v_{\mathsf{\overline{A}\,\overline{B}\,\overline{C}D}}, v_{\mathsf{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}}})^{\mathrm{T}} \end{array} \tag{2}$$

Using a standard technique from [5], the pair of linear fractional programs given by (1) and (2) is then transformed into the following equivalent pair of linear programs given by (3) and (4), assuming $q_C \geq 0$ $(C \in \{\mathsf{A}, \mathsf{AB}, \mathsf{ABC}, \mathsf{ABCD}, \mathsf{C}, \mathsf{CD}\})$, $x_A \geq 0$ $(A \in \mathcal{A}_{\mathcal{B}})$ and opt $\in \{\min, \max\}$. Note that for clarity of exposition only, we do not merge (3) and (4) by eliminating all $q_C$ with $C \in \{\mathsf{A}, \mathsf{AB}, \mathsf{ABC}, \mathsf{ABCD}, \mathsf{C}, \mathsf{CD}\}$.

$$
\begin{array}{l}
\text{opt } q_{\mathsf{ABCD}} \text{ subject to} \\
\begin{array}{lcl}
q_{\mathsf{A}} & = & 1 \\
q_{\mathsf{AB}} & \leq & 0.2 \cdot q_{\mathsf{A}} \\
q_{\mathsf{ABC}} & \leq & 0.2 \cdot q_{\mathsf{AB}} \\
q_{\mathsf{CD}} & \leq & 0.2 \cdot q_{\mathsf{C}}
\end{array}
\end{array}
\tag{3}
$$

$$
\begin{pmatrix} q_{\mathsf{A}} \\ q_{\mathsf{AB}} \\ q_{\mathsf{ABC}} \\ q_{\mathsf{ABCD}} \\ q_{\mathsf{C}} \\ q_{\mathsf{CD}} \end{pmatrix}
=
\begin{pmatrix}
1111\ 1111\ 0000\ 0000 \\
1111\ 0000\ 0000\ 0000 \\
1100\ 0000\ 0000\ 0000 \\
1000\ 0000\ 0000\ 0000 \\
1100\ 1100\ 1100\ 1100 \\
1000\ 1000\ 1000\ 1000
\end{pmatrix} \cdot
\begin{array}{l}
(x_{\mathsf{ABCD}}, x_{\mathsf{ABC\overline{D}}}, x_{\mathsf{AB\overline{C}D}}, x_{\mathsf{AB\overline{C}\,\overline{D}}}, \\
\ x_{\mathsf{A\overline{B}CD}}, x_{\mathsf{A\overline{B}C\overline{D}}}, x_{\mathsf{A\overline{B}\,\overline{C}D}}, x_{\mathsf{A\overline{B}\,\overline{C}\,\overline{D}}}, \\
\ x_{\mathsf{\overline{A}BCD}}, x_{\mathsf{\overline{A}BC\overline{D}}}, x_{\mathsf{\overline{A}B\overline{C}D}}, x_{\mathsf{\overline{A}B\overline{C}\,\overline{D}}}, \\
\ x_{\mathsf{\overline{A}\,\overline{B}CD}}, x_{\mathsf{\overline{A}\,\overline{B}C\overline{D}}}, x_{\mathsf{\overline{A}\,\overline{B}\,\overline{C}D}}, x_{\mathsf{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}}})^{\mathrm{T}}
\end{array}
\tag{4}
$$

The crux with this classical approach is that the number of atomic events and thus the number of variables of the generated linear programs is exponentiel in the number of basic events. Considering, for example, only twenty basic events, the generated linear programs already contain more than one million variables. Linear programs of this size generally cannot be tackled anymore by current computer technology and linear programming software.

The key idea of this paper now is to express the taxonomic relationships between the conjunctive events appearing in (3) in a different way. In detail, we suggest to substitute (4) by the following linear constraints:

$$
\begin{array}{l}
q_{\mathsf{A}} \geq q_{\mathsf{AB}} \geq q_{\mathsf{ABC}} \geq q_{\mathsf{ABCD}}, \\
q_{\mathsf{C}} \geq q_{\mathsf{ABC}} + q_{\mathsf{CD}} - q_{\mathsf{ABCD}},\ q_{\mathsf{CD}} \geq q_{\mathsf{ABCD}}\ .
\end{array}
\tag{5}
$$

We show that in our framework of probabilistic knowledge-bases and queries over conjunctive events, this idea yields linear programs for the computation of precise answers to probabilistic queries. The number of variables of these linear programs is given by the number of different conjunctive events in the probabilistic knowledge-base and the probabilistic query, increased by the number of additional variables necessary to express the constraints described by (5).

However, this new approach has still a major disadvantage over the classical one. We pay the reduced number of variables by an increased number of linear constraints. But also this problem can be tackled. We show that the linear programs given by (3) and (5) help us to generate a kind of necessary set of variables for the classical linear programming approach. The resulting linear programs of our example are then given by (3) and the following constraints with $y_C \geq 0$ $(C \in \{\mathsf{A\overline{B}\,\overline{C}\,\overline{D}}, \mathsf{AB\overline{C}\,\overline{D}}, \mathsf{ABC\overline{D}}, \mathsf{ABCD}, \mathsf{\overline{A}\,\overline{B}C\overline{D}}, \mathsf{\overline{A}\,\overline{B}CD}\})$:

$$
\begin{pmatrix} q_{\mathsf{A}} \\ q_{\mathsf{AB}} \\ q_{\mathsf{ABC}} \\ q_{\mathsf{ABCD}} \\ q_{\mathsf{C}} \\ q_{\mathsf{CD}} \end{pmatrix}
=
\begin{pmatrix}
111100 \\ 111000 \\ 110000 \\ 100000 \\ 110011 \\ 100010
\end{pmatrix} \cdot
\begin{pmatrix}
y_{\mathsf{ABCD}} \\ y_{\mathsf{ABC\overline{D}}} \\ y_{\mathsf{AB\overline{C}\,\overline{D}}} \\ y_{\mathsf{A\overline{B}\,\overline{C}\,\overline{D}}} \\ y_{\mathsf{\overline{A}\,\overline{B}CD}} \\ y_{\mathsf{\overline{A}\,\overline{B}C\overline{D}}}
\end{pmatrix}
\tag{6}
$$

We provide a classical deduction problem in which this new approach yields linear programs with a number of variables that is polynomial in the number of basic events.

Finally, let us also provide the result of our probabilistic deduction. The probabilistic query $?(\mathsf{BCD}|\mathsf{A})[x_1, x_2]$ to $(\mathcal{B}, \mathcal{P})$ yields the precise answer $[x_1, x_2] = [0, 0.04]$.

### 3.2 The Technical Details

In the sequel, let $(\mathcal{B}, \mathcal{P})$ be a probabilistic knowledge-base and let $?(F|E)[x_1, x_2]$ be a probabilistic query.

The classical approach (see e.g. [23] and [24]) exemplified in (3) and (4) is more formally expressed by the following two linear programs in (7) with $x_A \geq 0$ $(A \in \mathcal{A}_{\mathcal{B}})$ and opt $\in \{\min, \max\}$. Note that $\mathcal{P} \not\models E \to \bot$ implies the solvability of (7). In this case, the questioned precise bounds are given by the two optimal values. $\mathcal{P} \models E \to \bot$ implies the insolvability of (7).

$$
\begin{array}{l}
\qquad\qquad \text{opt } \sum_{A \in \mathcal{A}_{\mathcal{B}},\, A \preceq EF} x_A \text{ subject to} \\
\begin{array}{lcl}
\sum_{A \in \mathcal{A}_{\mathcal{B}},\, A \preceq E} x_A & = & 1 \\
\sum_{A \in \mathcal{A}_{\mathcal{B}},\, A \preceq GH} x_A & \geq & u_1 \cdot \sum_{A \in \mathcal{A}_{\mathcal{B}},\, A \preceq G} x_A \text{ and} \\
\sum_{A \in \mathcal{A}_{\mathcal{B}},\, A \preceq GH} x_A & \leq & u_2 \cdot \sum_{A \in \mathcal{A}_{\mathcal{B}},\, A \preceq G} x_A
\end{array} \\
\quad \text{for all } (H|G)[u_1, u_2] \in \mathcal{P}
\end{array}
\tag{7}
$$

To formalize the new approach introduced by (3) and (5), we need some preparations. We define the partial order $\sqsubseteq$ on $\mathcal{C}_{\mathcal{B}}$ by $C \sqsubseteq D$ iff $D \subseteq C$ for $C, D \in \mathcal{C}_{\mathcal{B}}$. The nonreflexive partial order $\sqsubset$ is canonically defined by $C \sqsubset D$ iff $C \sqsubseteq D$ and $C \neq D$ for $C, D \in \mathcal{C}_{\mathcal{B}}$. For $\mathcal{D} \subseteq \mathcal{C}_{\mathcal{B}}$, we define $\mathrm{Max}(\mathcal{D})$ to be the set of all maximal members of $\mathcal{D}$ with respect to $\sqsubseteq$.

The set of conjunctive events $\mathcal{C}$ in the probabilistic knowledge-base and the probabilistic query is defined by:

$$
\mathcal{C} = \bigcup \{\{G, GH\} \mid (H|G)[u_1, u_2] \in \mathcal{P}\} \cup \{E, EF\}\ . \tag{8}
$$

To determine the set of conjunctive events for which we must introduce a variable of the linear programs described by (3) and (5), we must increase $\mathcal{C}$ by certain conjunctive events, necessary to express the additional constraints exemplified in (5). The resulting set of conjunctive events $\mathcal{C}^{\circ}$ is the least set with $\mathcal{C} \subseteq \mathcal{C}^{\circ}$, and $C_1 C_2 \ldots C_k \in \mathcal{C}^{\circ}$ if $C \in \mathcal{C}^{\circ}$ and $C_1, C_2, \ldots, C_k \in \{C' \in \mathcal{C}^{\circ} \mid C' \sqsubset C\}$ with $k \geq 1$.

For $C_1, C_2, \ldots, C_k \in \mathcal{C}_{\mathcal{B}}$ with $k \geq 1$ let $q_{C_1, C_2, \ldots, C_k}$ be the abbreviation of

$$
\begin{array}{l}
\sum_{1 \leq i \leq k} q_{C_i} - \sum_{1 \leq i < j \leq k} q_{C_i C_j} + \\
\quad \sum_{1 \leq i < j < l \leq k} q_{C_i C_j C_l} - \cdots - (-1)^k \cdot q_{C_1 C_2 \ldots C_k}\ .
\end{array}
$$

Our new approach described by (3) and (5) is now formalized by the following two linear programs with $q_C \geq 0$ $(C \in \mathcal{C}^{\circ})$ and opt $\in \{\min, \max\}$:

$$
\begin{array}{l}
\qquad\qquad \text{opt } q_{EF} \text{ subject to} \\
\begin{array}{lcll}
q_E & = & 1 & \\
q_{\bot} & = & 0 & \text{if } \bot \in \mathcal{C}^{\circ} \\
q_{GH} & \geq & u_1 \cdot q_G & \text{for } (H|G)[u_1, u_2] \in \mathcal{P} \\
q_{GH} & \leq & u_2 \cdot q_G & \text{for } (H|G)[u_1, u_2] \in \mathcal{P} \\
q_C & \geq & q_{C_1, C_2, \ldots, C_k} & \text{for } C \in \mathcal{C}^{\circ},\ k \geq 1, \text{ and}
\end{array} \\
\quad \{C_1, C_2, \ldots, C_k\} = \mathrm{Max}(\{C' \in \mathcal{C}^{\circ} \mid C' \sqsubset C\})
\end{array}
\tag{9}
$$

Finally, formalizing our new approach exemplified in (3) and (6), we get the following two linear programs with $y_D \geq 0$ ($D \in \mathcal{C}^\circ$) and opt $\in \{\min, \max\}$. Note that the indices actually represent atomic instead of conjunctive events.

$$\begin{array}{lll} & \text{opt } \sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq EF} y_D \text{ subject to} & \\ \sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq E} y_D & = 1 & \\ y_\perp & = 0 \text{ if } \perp \in \mathcal{C}^\circ & \\ \sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq GH} y_D & \geq u_1 \cdot \sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq G} y_D \text{ and} & (10) \\ \sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq GH} y_D & \leq u_2 \cdot \sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq G} y_D & \\ \text{for all } (H|G)[u_1, u_2] \in \mathcal{P} & & \end{array}$$

The important next theorem now shows that the three presented linear programs in (7), (9) and (10) are equivalent. This means that $\mathcal{P} \not\models E \to \perp$ implies the solvability of (9) and (10). In this case, the questioned precise bounds are given by the two optimal values of (9) and equivalently by the two optimal values of (10). Furthermore, $\mathcal{P} \models E \to \perp$ implies the insolvability of (9) and (10).

**Theorem 3.1**

*a) For all solutions $(x_A)_{A \in \mathcal{A}_\mathcal{B}}$ of (7), there is a solution $(q_C)_{C \in \mathcal{C}^\circ}$ of (9) with:*

$$\sum_{A \in \mathcal{A}_\mathcal{B},\, A \preceq C} x_A = q_C \quad \text{for } C \in \mathcal{C}^\circ .$$

*b) For all solutions $(q_C)_{C \in \mathcal{C}^\circ}$ of (9), there is a solution $(y_D)_{D \in \mathcal{C}^\circ}$ of (10) with:*

$$q_C = \sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq C} y_D \quad \text{for } C \in \mathcal{C}^\circ .$$

*c) For all solutions $(y_D)_{D \in \mathcal{C}^\circ}$ of (10), there is a solution $(x_A)_{A \in \mathcal{A}_\mathcal{B}}$ of (7) with:*

$$\sum_{D \in \mathcal{C}^\circ,\, D \sqsubseteq C} y_D = \sum_{A \in \mathcal{A}_\mathcal{B},\, A \preceq C} x_A \quad \text{for } C \in \mathcal{C}^\circ .$$

**Proof.** We prove the three claims of this central theorem by providing the corresponding solutions. The fact that the presented values are well-defined, that they satisfy the indicated conditions and that they are indeed solutions of the corresponding linear programs can easily be verified.

a) Let $q_C = \sum_{A \in \mathcal{A}_\mathcal{B},\, A \preceq C} x_A$ for all $C \in \mathcal{C}^\circ$.

b) For $D \in \mathcal{C}^\circ$ let $y_D$ be defined by:

$$\begin{cases} q_D & \text{if } \{D' \in \mathcal{C}^\circ \mid D' \sqsubset D\} = \emptyset \\ q_D - q_{D_1, D_2, \ldots, D_k} & \text{if } \mathrm{Max}(\{D' \in \mathcal{C}^\circ \mid D' \sqsubset D\}) = \\ & \quad \{D_1, D_2, \ldots, D_k\} \text{ with } k \geq 1 . \end{cases}$$

c) Let $x_A = y_A$ for all $A \in \mathcal{A}_\mathcal{B} \cap \mathcal{C}^\circ$, and let $x_A = 0$ for all $A \in \mathcal{A}_\mathcal{B} \setminus \mathcal{C}^\circ$.

### 3.3 Short Summary

(10) presents a new linear programming technique for probabilistic deduction from probabilistic knowledge-bases over conjunctive events. Crucially, the number of variables of the generated linear programs is $|\mathcal{C}^\circ|$, and not $|\mathcal{A}_\mathcal{B}| = 2^{|\mathcal{B}|}$ like in the classical linear programming approach in (7).

To assure the usefulness of our new approach, we must show that $\mathcal{C}^\circ$ is often much smaller than $\mathcal{A}_\mathcal{B}$ and that it can be computed efficiently. In the next section, we provide an important classical deduction problem in which $|\mathcal{C}^\circ|$ is polynomial in the number of basic events. In Section 5, we provide an efficient algorithm for computing $\mathcal{C}^\circ$.

### 3.4 Examples

Let the probabilistic knowledge-base $(\mathcal{B}, \mathcal{P})$ be given by:

$$\begin{aligned} \mathcal{B} &= \{\mathsf{B}_i \mid i \in [1{:}6]\} , \\ \mathcal{P} &= \bigcup \{\{(\mathsf{B}_{i+1}|\mathsf{B}_i)[0, .2], (\mathsf{B}_i|\mathsf{B}_{i+1})[.9, 1]\} \mid i \in [1{:}5]\} . \end{aligned}$$

Let the probabilistic query be given by $?(\mathsf{B}_6|\mathsf{B}_1)[x_1, x_2]$.

We get $\mathcal{C} = \{\mathsf{B}_i \mid i \in [1:6]\} \cup \{\mathsf{B}_i\mathsf{B}_{i \bmod 6 + 1} \mid i \in [1:6]\}$ by (8). $\mathcal{C}$ is graphically represented by the grey nodes of Fig. 1. $\mathcal{C}^\circ$ is given by all the nodes of Fig. 1. Fig. 1 shows the Hasse-diagram of $(\mathcal{C}^\circ, \sqsubseteq)$. In this diagram, each node represents the conjunction of all basic events attached to the node itself and to all upper nodes. Thus, for instance, the node labeled with $\mathsf{B}_1$ represents the conjunctive event $\mathsf{B}_1$. The lowest node of the diagram represents $\mathsf{B}_1\mathsf{B}_2\mathsf{B}_3\mathsf{B}_4\mathsf{B}_5\mathsf{B}_6$.

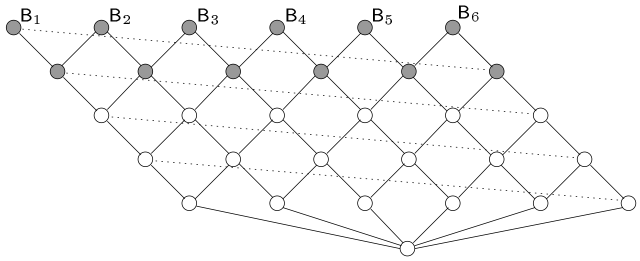


Figure 1: Hasse-diagram of $(\mathcal{C}^\circ, \sqsubseteq)$.

The linear programs of our new approach have only $|\mathcal{C}^\circ| = 31$ variables. The classical linear programs, in contrast, have $|\mathcal{A}_\mathcal{B}| = 64$ variables.

Computing the precise answer to the given probabilistic query is an instance of the classical problem of performing probabilistic deductions along a chain of $n \geq 3$ basic events (see e.g. [9], [1], [28], and [20]). In this more general problem, the number of variables of our new approach is only $|\mathcal{C}^\circ| = n^2 - n + 1$ . The number of variables of the classical approach, in contrast, is $|\mathcal{A}_\mathcal{B}| = 2^n$. Furthermore, in this classical problem, $\mathcal{C}^\circ$ is so well-structured that each of its members can be computed in time $O(1)$.

We get the important theoretical result that probabilistic deduction along a chain of basic events can be done in polynomial time in the length of the chain, since linear programming can be done in polynomial time in the size of the linear programs. From the practical point of view, our new approach enables us to perform probabilistic deductions along chains of even more than 100 basic events (we performed probabilistic deductions along a chain of 50 basic events in less than three minutes and along a chain of 100 basic events in less than one hour, on a HP 9000/725 workstation with public domain software for linear programming). The classical linear programming approach fails to solve problems of that size.

Many approaches of the literature iteratively apply inference rules for probabilistic deductions along a chain of basic events (see e.g. [1], [28], and [20]). Except for the fact that such an attempt may become a mathematical adventure (see especially [20]), it must be pointed out that all these approaches cannot be applied anymore if we just complete the considered chain to a cycle by adding two other probabilistic formulas. Our example could e.g. be modified by simply adding $\{(\mathsf{B}_1|\mathsf{B}_6)[.9,1], (\mathsf{B}_6|\mathsf{B}_1)[0,.0003]\}$ to $\mathcal{P}$. Our new linear programming technique, in contrast, can still be applied.

Our new approach can also be applied for probabilistic deduction along a chain of conjunctive events. Also this generalization cannot be tackled anymore by the iterative application of inference rules, since some of the conjunctive events in the chain might be taxonomically related (three taxonomically related conjunctive events are e.g. $\mathsf{B}_1\mathsf{B}_2$, $\mathsf{B}_2\mathsf{B}_3$ and $\mathsf{B}_1\mathsf{B}_3$).

Finally, let us also give a negative example to show the limits of our new approach. The set $\mathcal{C}^\circ$ cannot always be kept small. If we slightly change our example by moving our interest towards the probabilistic query $?(\mathsf{B}_6|\top)[x_1,x_2]$, we get $\mathcal{C} = \{\mathsf{B}_i \,|\, i \in [1:6]\} \cup \{\top\}$ and $\mathcal{C}^\circ = \mathcal{A}_\mathcal{B}$. In this extreme case, our new approach is not more efficient than the classical one.

In general, our new approach is more efficient than the classical one, if there are many maximal members in $\mathcal{C}$ or if the members of $\mathcal{C}$ contain many basic events (see also Section 4.4). The second condition is satisfied in many practical applications. However, the first condition rarely holds, since probabilistic knowledge-bases often contain probabilistic formulas with the true event in the premise.

The probabilistic queries $?(\mathsf{B}_6|\mathsf{B}_1)[x_1,x_2]$ and $?(\mathsf{B}_6|\top)[x_1,x_2]$ to the considered chain of basic events yield the precise answers $[x_1,x_2] = [0,.000542]$ and $[x_1,x_2] = [0,.000530]$, respectively. $?(\mathsf{B}_6|\mathsf{B}_1)[x_1,x_2]$ addressed to the considered cycle of basic events yields the precise answer $[x_1,x_2] = [0,.0003]$.

## 4 Exploiting Taxonomic Knowledge

In this section, we additionally consider taxonomic knowledge. Taxonomic knowledge can be exploited for an increased efficiency in both the classical and the new approach to probabilistic deduction.

### 4.1 The Key Ideas

Let the probabilistic knowledge-base of Section 3.1 be extended to $(\mathcal{B},\mathcal{T},\mathcal{P})$ by the additional taxonomic knowledge $\mathcal{T} = \{\mathsf{C} \rightarrow \mathsf{AB},\ \mathsf{D} \rightarrow \mathsf{C}\}$.

Additional taxonomic knowledge requires the falsity of certain atomic events (such atomic events are e.g. $\mathsf{AB\overline{C}D}$ and $\mathsf{A\overline{B}CD}$) and the corresponding variables in the classical approach described by (3) and (4) can be discarded in advance. The resulting linear programs of our example are then given by replacing (4) by the following constraints:

$$\begin{pmatrix} q_{\mathsf{A}} \\ q_{\mathsf{AB}} \\ q_{\mathsf{ABC}} \\ q_{\mathsf{ABCD}} \\ q_{\mathsf{C}} \\ q_{\mathsf{CD}} \end{pmatrix} = \begin{pmatrix} 111100 \\ 111000 \\ 110000 \\ 100000 \\ 110000 \\ 100000 \end{pmatrix} \cdot \begin{pmatrix} x_{\mathsf{ABCD}} \\ x_{\mathsf{ABC\overline{D}}} \\ x_{\mathsf{AB\overline{C}\,\overline{D}}} \\ x_{\mathsf{A\overline{B}\,\overline{C}\,\overline{D}}} \\ x_{\mathsf{\overline{A}B\overline{C}\,\overline{D}}} \\ x_{\mathsf{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}}} \end{pmatrix} \tag{11}$$

This idea of discarding variables in advance is already reported in the literature [31]. As a new contribution of this paper, we show that in our framework of probabilistic knowledge-bases and queries over conjunctive events, the variables (i.e. all not necessarily false atomic events) can be computed very efficiently.

Taxonomic knowledge can also be exploited in the new approach described by (3) and (5) to reduce the number of variables. Here, taxonomic knowledge requires syntactically different conjunctive events to have the same semantics under probabilistic interpretations (such conjunctive events are e.g. $\mathsf{C}$ and $\mathsf{ABC}$). We then use equivalence classes of conjunctive events as indices to the variables, instead of conjunctive events like in (3) and (5).

Again, the resulting linear programs help us to generate a kind of necessary set of variables for the classical linear programming approach under taxonomic knowledge described by (3) and (11). More precisely, (11) is replaced by the following constraints (here the indices $\mathsf{A}$, $\mathsf{AB}$, $\mathsf{ABC}$, $\mathsf{ABCD}$, $\mathsf{C}$ and $\mathsf{CD}$ on the left side are representatives of the corresponding equivalence classes):

$$\begin{pmatrix} q_{\mathsf{A}} \\ q_{\mathsf{AB}} \\ q_{\mathsf{ABC}} \\ q_{\mathsf{ABCD}} \\ q_{\mathsf{C}} \\ q_{\mathsf{CD}} \end{pmatrix} = \begin{pmatrix} 1111 \\ 1110 \\ 1100 \\ 1000 \\ 1100 \\ 1000 \end{pmatrix} \cdot \begin{pmatrix} y_{\mathsf{ABCD}} \\ y_{\mathsf{ABC\overline{D}}} \\ y_{\mathsf{AB\overline{C}\,\overline{D}}} \\ y_{\mathsf{A\overline{B}\,\overline{C}\,\overline{D}}} \end{pmatrix} \tag{12}$$

The probabilistic query $?(\mathsf{BCD}|\mathsf{A})[x_1,x_2]$ to $(\mathcal{B},\mathcal{T},\mathcal{P})$ yields the precise answer $[x_1,x_2] = [0, 0.008]$.

### 4.2 The Technical Details

Let $(\mathcal{B},\mathcal{T},\mathcal{P})$ be a taxonomic and probabilistic knowledge-base. Let $?(F|E)[x_1,x_2]$ be a probabilistic query and let $\mathcal{C}$ be defined like in (8) (i.e. $\mathcal{T}$ does not contribute to $\mathcal{C}$).

To formalize the exploitation of taxonomic knowledge in both the classical and the new approach, we define the hull-operator $\cdot^{\star}$ on $\mathcal{C}_\mathcal{B}$ as follows. Let the mapping $\mathcal{I}\colon \mathcal{C}_\mathcal{B} \longrightarrow \mathcal{C}_\mathcal{B}$ be defined by:

$$\mathcal{I}(C) = C \cup \bigcup\{N \mid M \rightarrow N \in \mathcal{T},\ M \subseteq C\} \text{ for all } C \in \mathcal{C}_\mathcal{B}\,.$$

Let the mapping $\cdot^{\star}\colon \mathcal{C}_\mathcal{B} \longrightarrow \mathcal{C}_\mathcal{B}$ be defined by $\cdot^{\star} = \mathcal{I}^k$ with $k \geq 0$ such that $\mathcal{I}^k = \mathcal{I}^{k+1}$ and $k$ minimal. It can easily be verified that $\cdot^{\star}$ is a hull-operator.

First, we examine the exploitation of taxonomic knowledge in the classical approach. There may be atomic events $A$ with $\mathcal{T} \models A \rightarrow \bot$. As already reported in [31], the variables corresponding to these atomic events can be discarded in advance in the classical approach in (7).

We show in [20] that the set of all remaining variables (i.e. the set of all not necessarily false atomic events) is given by $\mathcal{A}^{\star}_\mathcal{B} \setminus \{\bot\}$. This characterization yields a very efficient algorithm for the computation of all remaining variables, which is generally of exponential time complexity in the number of basic events. The details of this algorithm are given in Section 5.

Adapting the classical approach in (7) this way, we get the following two linear programs with $x_A \geq 0$ ($A \in \mathcal{A}^{\star}_\mathcal{B}$) and opt $\in \{\min, \max\}$:

$$\begin{array}{lcl} \multicolumn{3}{c}{\text{opt } \sum_{A \in \mathcal{A}^{\star}_\mathcal{B},\, A \preceq EF} x_A \text{ subject to}} \\ \sum_{A \in \mathcal{A}^{\star}_\mathcal{B},\, A \preceq E} x_A & = & 1 \\ x_\bot & = & 0 \text{ if } \bot \in \mathcal{A}^{\star}_\mathcal{B} \\ \sum_{A \in \mathcal{A}^{\star}_\mathcal{B},\, A \preceq GH} x_A & \geq & u_1 \cdot \sum_{A \in \mathcal{A}^{\star}_\mathcal{B},\, A \preceq G} x_A \text{ and} \\ \sum_{A \in \mathcal{A}^{\star}_\mathcal{B},\, A \preceq GH} x_A & \leq & u_2 \cdot \sum_{A \in \mathcal{A}^{\star}_\mathcal{B},\, A \preceq G} x_A \\ \multicolumn{3}{l}{\quad \text{for all } (H|G)[u_1,u_2] \in \mathcal{P}} \end{array} \tag{13}$$

Next, we examine how taxonomic knowledge can be exploited in our new approach in (9). There may be syntactically different conjunctive events that have the same semantics under probabilistic interpretations due to taxonomic knowledge. This observation yields a canonic equivalence relation on $\mathcal{C}_{\mathcal{B}}$ by $C \sim D$ iff $\mathcal{T} \models C \rightarrow D$ and $\mathcal{T} \models D \rightarrow C$ for all $C, D \in \mathcal{C}_{\mathcal{B}}$.

Thus, the variables of the linear programs in (9) are now related to equivalence classes of conjunctive events, and not to conjunctive events anymore. A representative of each equivalence class $[C]_{\sim}$ with $C \in \mathcal{C}_{\mathcal{B}}$ is given by $C^{\star}$.

The set of equivalence classes of conjunctive events for which we must introduce a variable is given as follows. The set of all equivalence classes of conjunctive events in the probabilistic knowledge-base and the probabilistic query is given by $\mathcal{C}^{\star}$. $\mathcal{C}^{\star}$ must be increased by certain equivalence classes, necessary to express additional constraints similar to (5). The resulting set of equivalence classes $\mathcal{C}^{\bullet}$ is the least set with $\mathcal{C}^{\star} \subseteq \mathcal{C}^{\bullet}$, and $(C_1 C_2 \dots C_k)^{\star} \in \mathcal{C}^{\bullet}$ if $C \in \mathcal{C}^{\bullet}$ and $C_1, C_2, \dots, C_k \in \{C' \in \mathcal{C}^{\bullet} \mid C' \sqsubset C\}$ with $k \geq 1$.

Modifying finally our new approach in (10) this way, we get the following two linear programs with $y_D \geq 0$ ($D \in \mathcal{C}^{\bullet}$) and opt $\in \{\min, \max\}$. Note that also here the indices actually represent atomic instead of conjunctive events.

$$\text{opt } \sum_{D \in \mathcal{C}^{\bullet},\, D \sqsubseteq EF} y_D \text{ subject to}$$

$$\begin{array}{lcl} \sum_{D \in \mathcal{C}^{\bullet},\, D \sqsubseteq E} y_D & = & 1 \\ y_{\perp} & = & 0 \text{ if } \perp \in \mathcal{C}^{\bullet} \\ \sum_{D \in \mathcal{C}^{\bullet},\, D \sqsubseteq GH} y_D & \geq & u_1 \cdot \sum_{D \in \mathcal{C}^{\bullet},\, D \sqsubseteq G} y_D \text{ and} \\ \sum_{D \in \mathcal{C}^{\bullet},\, D \sqsubseteq GH} y_D & \leq & u_2 \cdot \sum_{D \in \mathcal{C}^{\bullet},\, D \sqsubseteq G} y_D \\ \multicolumn{3}{l}{\quad \text{for all } (H|G)[u_1, u_2] \in \mathcal{P}} \end{array} \tag{14}$$

$\mathcal{T} \cup \mathcal{P} \not\models E \rightarrow \perp$ implies the solvability of (14). In this case, the questioned precise bounds are given by the two optimal values of (14). $\mathcal{T} \cup \mathcal{P} \models E \rightarrow \perp$ implies the insolvability of (14). These results can be proved similarly to the results of Section 3.2.

### 4.3 Short Summary

(13) presents a new characterization of the classical linear programming approach to probabilistic deduction under taxonomic knowledge. The number of variables is $|\mathcal{A}^{\star}_{\mathcal{B}}|$. (14) presents a new linear programming approach to probabilistic deduction under taxonomic knowledge. Crucially, the number of variables is $|\mathcal{C}^{\bullet}|$.

To justify our new approach in (14), it remains to show that $\mathcal{C}^{\bullet}$ is often much smaller than $\mathcal{A}^{\star}_{\mathcal{B}}$ and that it can be computed efficiently. The next section provides some examples illustrating the cardinalities of $\mathcal{C}^{\bullet}$ and $\mathcal{A}^{\star}_{\mathcal{B}}$. In Section 5, we present an efficient algorithm for computing $\mathcal{C}^{\bullet}$.

### 4.4 Examples

We consider the four taxonomic and probabilistic knowledge-bases defined by Fig. 2.1 to 2.4. For each figure, let $\mathcal{B}$ be the set of all characters appearing in the labels. Let $\mathcal{T}$ be the least set with $X \rightarrow Y \in \mathcal{T}$ for each edge from a node labeled with $X$ up to a node labeled with $Y$, and with $XY \rightarrow \perp \in \mathcal{T}$ for each pair of nodes labeled with $X$ and $Y$ contained in the same cycle. Let $\mathcal{P}$ be the least set with $(X|Y)[.2, .3] \in \mathcal{P}$ for each edge from a node labeled with $X$ up to a node labeled with $Y$. Let the probabilistic queries be given by ?(J|A)$[x_1, x_2]$ for Fig. 2.1, 2.3, and 2.4. Let the probabilistic query be given by ?(IJ|AB)$[x_1, x_2]$ for Fig. 2.2.

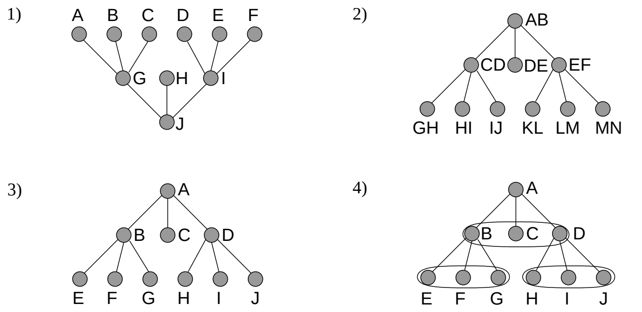


Figure 2: Taxonomic and probabilistic knowledge-bases.

A comparison of the cardinalities of $\mathcal{A}^{\star}_{\mathcal{B}}$ and $\mathcal{C}^{\bullet}$ is given by Table 1, left side. Generalizing the results, $\mathcal{C}^{\bullet}$ seems to be much smaller than $\mathcal{A}^{\star}_{\mathcal{B}}$ for taxonomic and probabilistic knowledge-bases such that $\mathcal{C}^{\star}$ contains many maximal members, and such that the members of $\mathcal{C}^{\star}$ contain many basic events. $\mathcal{A}^{\star}_{\mathcal{B}}$ and $\mathcal{C}^{\bullet}$ seem to have around the same cardinality for hierarchies of basic events and for hierarchies of pairwise disjoint basic events.

| Fig. | $\lvert\mathcal{A}^{\star}_{\mathcal{B}}\rvert$ | $\lvert\mathcal{C}^{\bullet}\rvert$ | $\lvert\mathcal{A}_{\mathcal{B}}\rvert$ | $\lvert\mathcal{C}^{\circ}\rvert$ |
|---|---|---|---|---|
| 2.1 | 163 | 10 | 1024 | 203 |
| 2.2 | 3136 | 79 | 16384 | 116 |
| 2.3 | 163 | 162 | 1024 | 203 |
| 2.4 | 11 | 11 | 1024 | 409 |

Table 1: The cardinalities of $\mathcal{A}^{\star}_{\mathcal{B}}$, $\mathcal{C}^{\bullet}$, $\mathcal{A}_{\mathcal{B}}$, and $\mathcal{C}^{\circ}$.

Since taxonomic knowledge can easily be expressed by probabilistic formulas, we could think to apply the purely probabilistic classical approach in (7) or even the purely probabilistic new approach presented in (10). The corresponding numbers of variables $|\mathcal{A}_{\mathcal{B}}|$ and $|\mathcal{C}^{\circ}|$ for the considered examples are given by Table 1, right side. A comparison to $|\mathcal{A}^{\star}_{\mathcal{B}}|$ and $|\mathcal{C}^{\bullet}|$ shows that it really makes sense to extract and to separate taxonomic from probabilistic knowledge.

The precise answer to all probabilistic queries is $[x_1, x_2] = [.04, .09]$.

## 5 Computing $\mathcal{A}^{\star}_{\mathcal{B}}$, $\mathcal{C}^{\bullet}$ and $\mathcal{C}^{\circ}$

We provide efficient algorithms for the computation of $\mathcal{A}^{\star}_{\mathcal{B}}$, $\mathcal{C}^{\bullet}$ and $\mathcal{C}^{\circ}$. In the sequel, let $(\mathcal{B}, \mathcal{T}, \mathcal{P})$ be a taxonomic and probabilistic knowledge-base. Let the probabilistic query be given by $?(F|E)[x_1, x_2]$ and let $\mathcal{C}$ be defined like in (8).

In all presented algorithms, we implicitly assume that conjunctive events are internally represented as bit vectors of length $|\mathcal{B}| + 1$ and that sets of conjunctive events are internally represented as balanced binary search trees.

The computation of $\mathcal{A}^{\star}_{\mathcal{B}}$ can be done by combining two well-known algorithms from the literature. The central role plays an algorithm from [12] that computes all the hulls of a given hull-operator. Assuming that the set of basic events $\mathcal{B}$ is given by $\{B_1, B_2, \dots, B_n\}$ with $n \geq 1$, this algorithm computes $\mathcal{A}^{\star}_{\mathcal{B}}$ as follows.

**Algorithm 5.1**

**function** $var1(\mathcal{B}, \mathcal{T})$;
**begin**
l1: $R := \emptyset^\star$;
$\mathcal{R} := \{R\}$;
**while** $R \cap \mathcal{B} \neq \mathcal{B}$ **do begin**
  $i := n$;
  **repeat**
    $i := i - 1$;
l2:   $H := ((R \cap \{B_1, B_2, \ldots, B_i\}) \,\dot\cup\, \{B_{i+1}\})^\star$
  **until** $B_{i+1} \notin R$ **and** $R \cap \{B_1, B_2, \ldots, B_i\} = H \cap \{B_1, B_2, \ldots, B_i\}$;
  $R := H$;
  $\mathcal{R} := \mathcal{R} \,\dot\cup\, \{R\}$
**end**;
**return** $\mathcal{R}$
**end**

The second algorithm is taken from [4]. It originally computes the hull of a set of attributes for a given set of functional dependencies in the area of database theory. Due to the equivalence of functional dependencies and taxonomic formulas [26], it can be applied to compute the hull $C^\star$ of single conjunctive events $C$ in (l1) and (l2). It runs in time $O(\|\mathcal{T}\|)$, where $\|\mathcal{T}\|$ denotes the input size of $\mathcal{T}$. Hence, Alg. 5.1 computes each member of $\mathcal{A}^\star_\mathcal{B}$ in time $O(|\mathcal{B}|^2 + |\mathcal{B}|\,\|\mathcal{T}\|)$.

The computation of $\mathcal{C}^\bullet$ for the case $|\mathrm{Max}(\mathcal{C}^\star)| = 1$ can be done in a similar way. Some probabilistic deduction problems with $|\mathrm{Max}(\mathcal{C}^\star)| = 1$ are given in Section 4.4 by the examples related to Fig. 2.2 to 2.4. The algorithm is based on the following lemma.

**Lemma 5.2** *$\mathcal{C}^\bullet$ is the least set with $\mathcal{C}^\star \subseteq \mathcal{C}^\bullet$, and $(C_1 C_2 \ldots C_k)^\star \in \mathcal{C}^\bullet$ if $C \in \mathrm{Max}(\mathcal{C}^\star)$ and $C_1, C_2, \ldots, C_k \in \{C' \in \mathcal{C}^\bullet \mid C' \sqsubset C\}$ with $k \geq 1$.*

**Proof.** For $C \in \mathcal{C}^\bullet$, there is $D \in \mathrm{Max}(\mathcal{C}^\bullet) = \mathrm{Max}(\mathcal{C}^\star)$ with $C \sqsubseteq D$ and thus $C_1, C_2, \ldots, C_k \in \{C' \in \mathcal{C}^\bullet \mid C' \sqsubset D\}$ for $C_1, C_2, \ldots, C_k \in \{C' \in \mathcal{C}^\bullet \mid C' \sqsubset C\}$. □

Lemma 5.2 yields $\mathcal{C}^\bullet = \{(\bigcup\{D \mid D \in \mathcal{D}\})^\star \mid \mathcal{D} \subseteq \mathcal{C}^\star\}$ for the case $|\mathrm{Max}(\mathcal{C}^\star)| = 1$. We define a canonic hull-operator $\cdot^{\circledast}$ on the set of all subsets of $\mathcal{C}^\star$ by:

$$\mathcal{D}^{\circledast} = \{C \in \mathcal{C}^\star \mid (\bigcup\{D \mid D \in \mathcal{D}\})^\star \sqsubseteq C\} \ \text{ for all } \mathcal{D} \subseteq \mathcal{C}^\star .$$

We get $\mathcal{C}^\bullet = \{\bigcup \mathcal{D}^{\circledast} \mid \mathcal{D} \subseteq \mathcal{C}^\star\} = \{\bigcup \mathcal{E} \mid \mathcal{E} \in (2^{\mathcal{C}^\star})^{\circledast}\}$. Thus, it remains to compute $(2^{\mathcal{C}^\star})^{\circledast}$. This can be done by a slight modification of Alg. 5.1, replacing $\mathcal{B}$ by $\mathcal{C}^\star$ and $\cdot^\star$ by $\cdot^{\circledast}$. Since $\cdot^{\circledast}$ is computable in time $O(\|\mathcal{T}\| + |\mathcal{B}|\,|\mathcal{C}^\star|)$, the modification of Alg. 5.1 computes each member of $\mathcal{C}^\bullet$ in time $O(\|\mathcal{T}\|\,|\mathcal{C}^\star| + |\mathcal{B}|\,|\mathcal{C}^\star|^2)$.

The computation of $\mathcal{C}^\circ$ for the case $|\mathrm{Max}(\mathcal{C})| = 1$ can be done in the same way, since $\mathcal{C}^\bullet = \mathcal{C}^\circ$ and $\mathcal{C}^\star = \mathcal{C}$ for $\mathcal{T} = \emptyset$. Hence, each member of $\mathcal{C}^\circ$ is computable in time $O(|\mathcal{B}|\,|\mathcal{C}|^2)$ for $|\mathrm{Max}(\mathcal{C})| = 1$.

The efficient computation of $\mathcal{C}^\bullet$ for the case $|\mathrm{Max}(\mathcal{C}^\star)| > 1$ requires a completely different technique. A probabilistic deduction problem with $|\mathrm{Max}(\mathcal{C}^\star)| > 1$ is given in Section 4.4 by the example related to Fig. 2.1. The algorithm is based on the next lemma.

**Lemma 5.3** *$\mathcal{C}^\bullet$ is the least set with $\mathcal{C}^\star \subseteq \mathcal{C}^\bullet$, and $(C_1 C_2)^\star \in \mathcal{C}^\bullet$ if $C \in \mathrm{Max}(\mathcal{C}^\star)$ and $C_1, C_2 \in \{C' \in \mathcal{C}^\bullet \mid C' \sqsubset C\}$.*

**Proof.** The claim follows from Lemma 5.2 by induction on the number $k \geq 1$ of conjunctive events that appear in the conjunction $(C_1 C_2 \ldots C_k)^\star$. □

The algorithm is given as follows. It exploits the idea of semi-naive fixed point evaluation (see especially [30]) from the area of deductive databases for an increased efficiency.

**Algorithm 5.4**

**function** $var2(\mathcal{T}, \mathcal{C})$;
**begin**
$\mathcal{H} := \mathcal{C}^\star$;
$\mathcal{M} := \mathrm{Max}(\mathcal{H})$;
$(\mathcal{S}_M)_{M \in \mathcal{M}} := (\emptyset)_{M \in \mathcal{M}}$;
**repeat**
  $(\triangle_M)_{M \in \mathcal{M}} := (\{C \in \mathcal{H} \mid C \sqsubset M\} \setminus \mathcal{S}_M)_{M \in \mathcal{M}}$;
  $(\mathcal{S}_M)_{M \in \mathcal{M}} := (\mathcal{S}_M \,\dot\cup\, \triangle_M)_{M \in \mathcal{M}}$;
l1: $\mathcal{H} := \{(C_1 C_2)^\star \mid \exists\, M \in \mathcal{M} : C_1 \in \triangle_M,\ C_2 \in \mathcal{S}_M\}$
**until** $\mathcal{H} = \emptyset$;
**return** $\bigcup_{M \in \mathcal{M}} \mathcal{S}_M$
**end**

To determine the time complexity of Alg. 5.4, let us analyze more deeply (l1). $\mathcal{H}$ and $\mathcal{S}_M$ for all $M \in \mathcal{M}$ are contained in $\mathcal{C}^\bullet$. Each time (l1) is executed, it runs in time $O((|\mathcal{B}| + \|\mathcal{T}\| + log_2|\mathcal{C}^\bullet|)\,|\mathcal{M}|\,|\bigcup_{M \in \mathcal{M}} \triangle_M|\,|\mathcal{C}^\bullet|)$ and thus in time $O((|\mathcal{B}| + \|\mathcal{T}\|)\,|\mathcal{M}|\,|\bigcup_{M \in \mathcal{M}} \triangle_M|\,|\mathcal{C}^\bullet|)$, since $|\mathcal{C}^\bullet| \leq 2^{|\mathcal{B}|+1}$. Hence, Alg. 5.4 computes each member of $\mathcal{C}^\bullet$ in time $O((|\mathcal{B}| + \|\mathcal{T}\|)\,|\mathrm{Max}(\mathcal{C}^\star)|\,|\mathcal{C}^\bullet|)$.

Again, the computation of $\mathcal{C}^\circ$ for the case $|\mathrm{Max}(\mathcal{C})| > 1$ can be done in the same way. Hence, each member of $\mathcal{C}^\circ$ can be computed in time $O(|\mathcal{B}|\,|\mathrm{Max}(\mathcal{C})|\,|\mathcal{C}^\circ|)$ for $|\mathrm{Max}(\mathcal{C})| > 1$.

## 6 Extending the Language

The considered language for representing taxonomic and probabilistic knowledge is quite scarce. However, it can be extended easily without influencing the presented results. We could additionally consider taxonomic formulas $C \not\rightarrow D$ with $C, D \in \mathcal{C}_\mathcal{B}$ of the following semantics:

$$\mathrm{P} \models C \not\rightarrow D \ \text{ iff } \ \mathrm{P}(C) \neq \mathrm{P}(CD)\,.$$

We show in [20] that they can be added to the taxonomic and probabilistic knowledge-base without affecting the results of probabilistic queries as far as the taxonomic and probabilistic knowledge-base remains satisfiable.

Furthermore, we could generalize our approach to probabilistic formulas $r_1 C_1 + r_2 C_2 + \cdots + r_k C_k \geq 0$ with $k \geq 1$, $r_i \in \mathbf{R}$ and $C_i \in \mathcal{C}_\mathcal{B}$ for all $i \in [1\!:\!k]$ adapted from [10] of the following semantics:

$$\mathrm{P} \models r_1 C_1 + r_2 C_2 + \cdots + r_k C_k \geq 0 \text{ iff}$$
$$r_1 \mathrm{P}(C_1) + r_2 \mathrm{P}(C_2) + \cdots + r_k \mathrm{P}(C_k) \geq 0\,.$$

We could also consider probabilistic formulas $I(A, C, B)$ with $A, B, C \in \mathcal{C}_\mathcal{B}$ expressing the conditional independence of $A$ and $B$, given $C$ (see especially [25]):

$$\mathrm{P} \models I(A, C, B) \ \text{ iff } \ \mathrm{P}(ABC) \cdot \mathrm{P}(C) = \mathrm{P}(AC) \cdot \mathrm{P}(BC)\,.$$

In both cases, we just have to adapt the definition of $\mathcal{C}$ in (8) to the new probabilistic formulas. Of course, if probabilistic formulas for representing conditional independence are involved, precise answers to probabilistic queries must be computed by nonlinear programming.

## 7 Summary and Outlook

We presented a new, efficient linear programming approach to probabilistic deduction from probabilistic knowledge-bases over conjunctive events. We showed that our new approach solves the classical problem of probabilistic deduction along a chain of basic events in polynomial time in the length of the chain.

We then elaborated the exploitation of taxonomic knowledge in our new linear programming approach for an increased efficiency. We also provided new results for the classical linear programming approach to probabilistic deduction under taxonomic knowledge.

Further research could concern the interesting question if and how our new linear programming approach can be applied for realizing an efficient subset of first-order probabilistic deduction (see especially [3] and [14]).

## References


[1] S. Amarger, D. Dubois, and H. Prade. Constraint propagation with imprecise conditional probabilities. In B. D'Ambrosio, P. Smets, and P. P. Bonissone, editors, *Proc. of the $7^{th}$ Conference on Uncertainty in Artificial Intelligence*, pages 26–34. Morgan Kaufmann Publishers, 1991.

[2] K. A. Andersen and J. N. Hooker. Bayesian logic. *Decision Support Systems*, 11:191–210, 1994.

[3] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge: a logical approach.* The MIT Press, Cambridge, USA, 1990.

[4] C. Beeri and P. A. Bernstein. Computational problems related to the design of normal form relational schemas. *ACM Transactions on Database Systems*, 4:30–59, 1979.

[5] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quaterly*, 9:181–186, 1962.

[6] V. Chvátal. *Linear Programming.* Freeman, San Francisco, 1983.

[7] B. de Finetti. *Theory of Probability.* Wiley, New York, 1974.

[8] D. Dubois and H. Prade. On fuzzy syllogisms. *Computational Intelligence*, 4(2):171–179, 1988.

[9] D. Dubois, H. Prade, and J.-M. Touscas. Inference with imprecise numerical quantifiers. In Z. W. Ras and M. Zemankova, editors, *Intelligent Systems*, chapter 3, pages 53–72. Ellis Horwood, 1990.

[10] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87:78–128, 1990.

[11] A. M. Frisch and P. Haddawy. Anytime deduction for probabilistic logic. *Artificial Intelligence*, 69:93–122, 1994.

[12] B. Ganter. Algorithmen zur Formalen Begriffsanalyse. In B. Ganter, R. Wille, and K. E. Wolff, editors, *Beiträge zur Begriffsanalyse*, pages 241–254. B.I.-Wissenschaftsverlag, Mannheim, 1987.

[13] G. Georgakopoulos, D. Kavvadias, and C. H. Papadimitriou. Probabilistic satisfiability. *Journal of Complexity*, 4(1):1–11, 1988.

[14] J. Y. Halpern. An analysis of first-order logics of probability. *Artifical Intelligence*, 46:311–350, 1990.

[15] P. Hansen, B. Jaumard, G.-B. D. Nguetsé, and M. P. de Aragão. Models and algorithms for probabilistic and Bayesian logic. In C. S. Mellish, editor, *Proc. of the $14^{th}$ International Joint Conference on Artificial Intelligence*, pages 1862–1868, 1995.

[16] J. Heinsohn. Probabilistic description logics. In R. L. de Mantaras and D. Poole, editors, *Proc. of the $10^{th}$ Conference on Uncertainty in Artificial Intelligence.* Morgan Kaufmann Publishers, 1994.

[17] D. Kavvadias and C. H. Papadimitriou. A linear programming approach to reasoning about probabilities. *Annals of Mathematics and Artificial Intelligence*, 1:189–205, 1990.

[18] R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge Based Systems.* Artificial Intelligence. Springer Verlag, 1991.

[19] T. Lukasiewicz. Uncertain reasoning in concept lattices. In *Proc. of the $3^{rd}$ European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 946 of *Lecture Notes in Artificial Intelligence*, pages 293–300, 1995.

[20] T. Lukasiewicz. *Precision of Probabilistic Deduction under Taxonomic Knowledge.* PhD thesis, Universität Augsburg, May 1996.

[21] T. Lukasiewicz, W. Kießling, G. Köstler, and U. Güntzer. Taxonomic and uncertain integrity constraints in object-oriented databases - the TOP approach. In N. Pissinou, A. Silberschatz, E. K. Park, and K. Makki, editors, *Proc. of the $4^{th}$ International Conference on Information and Knowledge Management*, pages 241–249, Baltimore, Maryland, USA, Nov. 1995.

[22] C. Luo, C. Yu, J. Lobo, G. Wang, and T. Pham. Computation of best bounds of probabilities from uncertain data. *Computational Intelligence*, 12(4):541–566, 1996.

[23] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–88, 1986.

[24] G. Paaß. Probabilistic Logic. In D. Dubois, P. Smets, A. Mamdani, and H. Prade, editors, *Non-Standard Logics for Automated Reasoning*, chapter 8, pages 213–251. Academic Press, London, 1988.

[25] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, California, 1988.

[26] Y. Sagiv, C. Delobel, D. S. Parker, and R. Fagin. An equivalence between relational database dependencies and a fragment of propositional logic. *Journal of the Association for Computer Machinery*, 28(3):435–453, 1981.

[27] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, New York, 1986.

[28] H. Thöne. *Precise Conclusion under Uncertainty and Incompleteness in Deductive Database Systems.* PhD thesis, Universität Tübingen, July 1994.

[29] H. Thöne, U. Güntzer, and W. Kießling. Towards precision of probabilistic bounds propagation. In D. Dubois, M. P. Wellman, B. D. Ambrosio, and P. Smets, editors, *Proc. of the $8^{th}$ Conference on Uncertainty in Artificial Intelligence*, pages 315–322. Morgan Kaufmann Publishers, July 1992.

[30] J. D. Ullman. *Principles of Database and Knowledge-based Systems, Vol. 1.* Computer Science Press, 1989.

[31] M. von Rimscha. The determination of comparative and lower probability. In *Workshop Uncertainty in Knowledge-Based Systems, FAW-B-90025*, volume 2, pages 344–376, Ulm, Germany, 1990.