Online Appendix to: Indeterministic Handling of Uncertain Decisions in Deduplication

FABIAN PANSE, University of Hamburg, Germany

MAURICE VAN KEULEN, University of Twente, the Netherlands

NORBERT RITTER, University of Hamburg, Germany

A. DEMONSTRATING EXAMPLE FOR SOME DISCUSSION ON RELATED WORK

As mentioned in Section 2, Ioannou et al. cannot ensure that from their approach always the most probable worlds result. Moreover, this approach disregards negative information (information that two database entities are nonduplicates for sure), which makes the result more inaccurate and which makes an introduction of negative domain knowledge during the initially offline performed linkage creation impossible.

As an example, consider the probabilistic linkage database $PLDB = \langle \mathcal{E}, \mathcal{L}, p^a, p^l \rangle$ with the five entities $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5\}$, which has been already used in Ioannou et al. [2010], but let us assume that the linkage between e_4 and e_5 has a probability of $p^l(e_4, e_5) = 0.4$ instead of 0.8 and let us assume that e_2 and e_3 are not linked because a domain expert know with absolute certainty that both entities are no duplicates ($\Rightarrow p^l(e_2, e_3) = 0$). The modified database is graphically illustrated in Figure 16(i).

We understand the approach of Ioannou et al. in a way that linkages between some entities, for example, the linkage between e_3 and e_4 , are not present in Figure 1 in Ioannou et al. [2010], because their probabilities (e.g., $p^l(e_3, e_4)$) are too small to be of interest. As a consequence, the probabilities of all linkages (except the linkage between e_2 and e_3) that are not presented in this example can be greater than 0. Thus, we assume that all linkages presented in Figure 1 in Ioannou et al. [2010] result from using a threshold T_l that demarcates all the linkages being of interest (all linkages with a probability greater than or equal to T_l) from all remaining linkages. For instance, our modified example (Figure 16(i)) could result by using a threshold $T_l = 0.4$, because the lowest probability of all presented linkages is 0.4. The probabilities of all missing linkages must be smaller than 0.4, because otherwise (for reasons of consistency) these linkages must be present in this figure as well. By increasing T_l from 0.4 to 0.8, the linkage between e_1 and e_3 as well as the linkage between e_4 and e_5 are further removed (see Figure 16(ii)).

Let Q be a query that has a condition that is satisfied by all five entities. Thus, from applying Q with the threshold $T_l = 0.8$ (Figure 16(ii)), the two possible worlds $I_1 = \{e_1, e_2, e_3, e_4, e_5\}$ and $I_2 = \{\mu(\{e_1, e_2\}), e_3, e_4, e_5\}$ result. However, since the case of nonexistence for the linkage $p^l(e_1, e_2)$ is more unlikely than the case of existence of the linkage $p^l(e_4, e_5)$, the ignored possible world $I_3 = \{\mu(\{e_1, e_2\}), e_3, \mu(\{e_4, e_5\})\}$ is more probable than the considered world I_1 . For that reason, in our (α, β) -restrictions, we always restrict the decision considered to be uncertain at both ends of the probability range.

Furthermore, let the threshold be $T_l = 0.4$. As mentioned previously, this configuration leads to all the linkages explicitly presented in Figure 16(i). Because, for each linkage specification $\mathcal{L}^{sp} \supset \mathcal{L}$, the transitive closure is performed (among others), the possible world $I_4 = \{\mu(\{e_1, e_2, e_3\}), e_4, e_5\}$ results (compare with the possible world I_2 in Example 2 in Ioannou et al. [2010]), Nevertheless, since e_2 and e_3 are not linked, because both entities are known to be definitely nonduplicates, this world is definitely

Fig. 16. The marginally modified sample probabilistic linkage database from Figure 1 in Ioannou et al. [2010], with the two threshold settings (i) $T_l = 0.4$ and (ii) $T_l = 0.8$.

not possible. As a consequence, information on nonduplicates is ignored during query evaluation and hence cannot be incorporated by the user (or domain expert) during linkage creation. In order to avoid such situations, we firstly consider all matching results (whether possible or not), then create all possible combination of uncertain edges (worlds) and finally remove impossible ones (see Section 5.1.2).

B. INDETERMINISTIC DEDUPLICATION WITH MAYBMS

MayBMS is a probabilistic database management system developed at the Cornell University [Koch 2008, 2009]. For a succinct representation of probabilistic data with attribute value dependencies and tuple correlations MayBMS uses the concept of U-relations (U-relational databases), which is based on probabilistic conditional tables [Suciu et al. 2011]. This means that tuples can be assigned with Boolean conditions defined on a finite set of variables.⁸ Each variable can be interpreted as an independent probabilistic event. U-relational databases are restricted to a finite set of possible worlds. Thus, each variable is defined on a finite domain (its set of possible outcomes). Correlations are modeled by using same variables for the values of different attributes (attribute value dependencies) or for different tuples (tuple correlations). To increase succinctness further on, MayBMS uses vertical partitioning to decompose tuples in several probabilistic independent set of attributes (also denoted as normalization [Suciu et al. 2011]). A U-relation consists of a set of attributes to be stored (*value columns*) and, in addition, a set of pairs (V_i, D_i) (*condition columns*) for modeling variable assignments that serves as atomic conditions for tuple existence. The possible outcomes of variables along with their probabilities are separately stored in a ternary relation $\mathbf{W}$, called *world-table*. As ULDB, U-relational databases are a complete and closed representation system for finite probabilistic databases. From a U-relational database, an arbitrary possible world can be derived by (1) mapping each variable in $\mathbf{W}$ to one of its possible outcomes (the world's corresponding variable assignment θ), (2) joining each U-relation with $\mathbf{W}$ on the condition columns, (3) selecting all tuples whose conditions satisfy θ , and (4) projecting away the condition columns.

⁸Note, in U-relational databases variables are not allowed as attribute values [Koch 2009].

$U_{\mathcal{R}_1[\text{name}]}$	V	D	TID	name
	v	1	t_1	Tim
	v	2	t_1	Jim
	w	1	t_2	Tim

$U_{\mathcal{R}_1[\text{firm}]}$	V	D	TID	firm
	v	1	t_1	Oracle
	v	2	t_1	Nokia
	w	1	t_2	IBM

$U_{\mathcal{R}_2}$	V	D	TID	firm	industry
	x	1	t_3	IBM	software
	y	1	t_4	Oracle	software
	z	1	t_5	Nokia	cell-phone

W	V	D	P
	v	1	0.3
	v	2	0.7
	w	1	0.6
	x	1	1.0
	y	1	1.0
	z	1	1.0

Fig. 17. Succinct representation of the sample probabilistic database of Figure 3 with U-relational databases.

$U_{\mathcal{R}[\text{corr.}]}$	V	D	TID
	x	1	t_1
	x	1	t_2
	x	2	t_{12}

W	V	D	P
	x	1	0.4
	x	2	0.6

Fig. 18. Modeling $\text{cpl}(\{\{t_1, t_2\}, \{t_{12}\}\})$ with U-relational databases.

A succinct U-relational database representation whose set of possible worlds is equivalent to this of the ULDB database shown in Figure 3 is presented in Figure 17.

B.1. Modeling Tuple Dependencies in MayBMS

Since U-relational databases are principally based on the concept of conditional tables, tuple dependencies can be simply modeled by variables. To model the complementation $\text{cpl}(\{A_1, \dots, A_k\})$, a new U-relation $U_{\mathcal{R}[\text{corr.}]}$ and one variable x with k possible outcomes is required. The newly created U-relation $U_{\mathcal{R}[\text{corr.}]}$ needs only one pair of condition columns, and hence has the following three attributes: (1) **V** for storing the variable x , (2) **D** for storing the possible outcome of x being the condition that the considered tuple exists, and (3) **TID** for storing the identifier of the considered tuple. Each tuple $t \in \bigcup_{i \in \{1, \dots, k\}} A_i$ is stored in $U_{\mathcal{R}[\text{corr.}]}$ with the same variable $\mathbf{V} = x$. To each tuple in A_i the possible outcome i is assigned. Finally, the newly created variable x along with the probabilities of its k possible outcomes are stored in the world-table **W**.

For demonstration, we consider the same example as already used in Section 3.2.2, that is, the complementation $\text{cpl}(\{\{t_1, t_2\}, \{t_{12}\}\})$. This dependency is modeled by creating the U-relation $U_{\mathcal{R}[\text{corr.}]}$ and extending the world table **W** as illustrated in Figure 18.

B.2. Generation of Probabilistic Result Data with U-Relations

For representing the set of possible worlds $W = \{I_1, \dots, I_k\}$ resulting from an indeterministic deduplication as a U-relational database, a new variable x of the world-table **W** having $|W|$ possible outcomes (one for each world) is required. For each tuple t of the possible world I_j , a new tuple (x, j, t) is inserted into the U-relation $U_{\mathcal{R}[\text{corr.}]}$, where j is the index number of the considered possible world.

For the purpose of demonstration, we consider our example already used for x-relation generation. We create a variable x with one possible value for each of the five resultant possible worlds $W = \{I_1, I_2, I_3, I_4, I_8\}$ and generate the new tuples in the way

$U_{\mathcal{R}[\text{corr.}]}$	V	D	TID
	x	1	t_1
	x	1	t_2
	x	1	t_3
	x	2	t_{12}
	x	2	t_3
	x	3	t_{13}
	x	3	t_2
	x	4	t_{23}
	x	4	t_1
	x	5	t_{123}

W	V	D	P
	x	1	0.138
	x	2	0.553
	x	3	0.092
	x	4	0.059
	x	5	0.158

Fig. 19. U-relation $U_{\mathcal{R}[\text{corr.}]}$ (left) and world-table **W** (right).

Input: Set of consistent W-graphs $WSet$	
1.	$W = \emptyset$
2.	For each graph $G = (N, E, P) \in WSet$
2.1	$I = \emptyset$
2.3	For each component $G_i = (N_i, E_i)$
	$I = I \cup \{\mu(N_i)\}$
2.4	$W = W \cup \{I\}$
2.5	$P(I) = P$
Output: Set of possible worlds $W = \{I_1, I_2, \dots, I_k\}$, Probability distribution $P : W \rightarrow (0, 1]$	

Fig. 20. Algorithm for possible world creation.

described previously (prior condition variables are assumed to be nonexistent). The resultant U-relation $U_{\mathcal{R}[\text{corr.}]}$ and the extended world-table **W** are shown in Figure 19.

C. ALGORITHM

In this section, first we present an algorithm for possible world creation and then two algorithms for generating probabilistic result data (one for the ULDB model and one for U-relational databases).

C.1. Algorithm for Possible World Creation

An algorithm for possible world creation is shown in Figure 20. The input of the algorithm is a set of consistent W -graphs ($WSet$). Based on these W -graphs, a set of possible worlds (denoted as W) is generated (one world for each consistent W -graph). For each W -graph, an initially empty world I is defined (Step 2.1). Then, for each of the W -graph's components, a tuple is added to this possible world by merging the tuples belonging to the component's nodes (Step 2.2). Finally, the resultant world is added to the set of possible worlds (Step 2.3) and its probability is defined as the probability of the corresponding W -graph (Step 2.4).

C.2. Algorithm for Generating X-Relations

A complete algorithm for x-relation generation is shown in Figure 21. The input of the algorithm is W , a set of possible worlds and P a probability distribution over

Fig. 21. Algorithm for x-relation generation.

these worlds.⁹ First, a new indicator tuple is created (Step 1). Second, for each possible world, an alternative of the indicator tuple is generated (Step 2.1). Then, we iterate over all tuples of the considered world (Step 2.2). If a tuple already belongs to the output x-relation $\mathcal{R}_X$, the lineage and probability of this tuple is adapted. Otherwise, the tuple along with its new lineage and probability is inserted into $\mathcal{R}_X$. Finally (Step 3), prior lineage (the condition λ') is taken into account. For merged tuples, we consider the prior lineage generation as a part of the tuple merging step.

C.3. Algorithm for Generating a U-relational Database

A complete algorithm for U-relation generation is shown in Figure 22. The input of the algorithm is a set of possible worlds $W = \{I_1, I_2, \dots, I_k\}$. First, the variable x is created (Step 1). Second, for each possible world I_j the tuple $(x, j, P(I_j))$ is inserted in the world-table $\mathbf{W}$ (Step 2.1). Finally, for each tuple $t \in I_j$, the tuple $(x, j, t.TID)$ ¹⁰ is inserted into the resultant U-relation $U_{\mathcal{R}[\text{corr}]}$ (Step 2.2).

D. COMPLEXITY

In this section, we formally discuss the complexity of our full-indeterministic approach. As known from other techniques, which are based on the possible world semantics, a full-indeterministic approach has, in theory, a very high complexity. The number of W -graphs that can be generated from an M -graph with k uncertain edges is:

$$N_{W\text{-graph}}(k) = 2^k.$$

⁹With M -graph decomposition, we have a set of world-sets along with their probability distributions (one set and one distribution per *partial M-graph*). In that case, the algorithm is applied to each world-set individually.

¹⁰We assume that each tuple has a unique tuple-identifier TID.

<p>Input: Set of possible worlds $W = \{I_1, I_2, \dots, I_k\}$</p> <p>Probability distribution $P : W \rightarrow (0, 1]$</p> <ol style="list-style-type: none"> 1. Create a variable x 2. For each world $I_j \in W$ <ol style="list-style-type: none"> 2.1 Insert the tuple $(x, j, P(I_j))$ into W 2.2 For each tuple $t \in I_j$ <ol style="list-style-type: none"> Insert the tuple $(x, j, t.TID)$ into $U_{\mathcal{R}[\text{corr}]}$ <p>Output: U-relation $U_{\mathcal{R}[\text{corr}]}$, world table W</p>

Fig. 22. Algorithm for U-relation generation.

A fully connected M -graph with n nodes has $|E| = n(n-1)/2$ edges. As a consequence, given a source relation with n tuples, the number of resultant W -graphs ($k = |E|$) is at most:

$$N_{W\text{-graph}}^{\max} = N_{W\text{-graph}}(|E|) = 2^{n(n-1)/2}.$$

The number of consistent possible worlds that can result from a certain source relation with n tuples, where each duplicate decision is uncertain, is equal to the number of possible partitions of the relation's tuples. Thus, the maximal possible number of resultant possible worlds can be reduced to the complexity of set partitioning and is:

$$N_{PW}^{\max} = B_n = \frac{1}{e} \sum_{i=1}^{\infty} \frac{i^n}{i!},$$

where B_n is the n th bell number [Rota 1964].

If each edge is uncertain, the resultant x-relation can be mapped to the power set of the source relation's tuples without the empty set ($2^{\mathcal{R}} \setminus \emptyset$). Thus, in the worst case, the number of resultant x-tuples is:

$$N_{|\mathcal{R}_X|}^{\max} = |2^{\mathcal{R}}| - 1 = 2^{|\mathcal{R}|} - 1.$$

In order to get an idea of the dramatic complexity scale, we assume a source relation $\mathcal{R}$ with 10 tuples. The number of W -graphs that can be generated from the initial M -graph is maximal:

$$N_{W\text{-graph}}^{\max} = 2^{45} \simeq 3.5184 \cdot 10^{13}.$$

The number of resultant possible worlds and hence the number of indicator alternatives is at most:

$$N_{PW}^{\max} = B_{10} = 115,975.$$

Finally, the maximal number of resultant x-tuples is:

$$N_{|\mathcal{R}_X|}^{\max} = 2^{10} - 1 = 1,023 \simeq 100 \cdot |\mathcal{R}|.$$

In summary, the complexity of the indeterministic deduplication algorithm, as well as the size of the resultant data increases dramatically with the number of uncertain edges. As a consequence, a full-indeterministic approach is in general only of theoretical value and not usable in practice.

E. EQUIVALENT AND WORKING QUERIES

Although we follow the TriQL manual, the queries Q1-Q3 presented in Section 6 are only of a theoretical nature and do not work on the currently available version of the

Trio prototype¹¹ (query *Q4* works as it is). In this section, we present sequences of TriQL commands that are working on the current Trio version and that lead to same results as the presented ones.

E.1. Working Implementation of Query *Q1*

Query *Q1* does not work, because it uses the IN-predicate and a vertical subquery. The IN-predicate can be realized by a join and the subquery can be realized by an auxiliary table *MaxIndAlternative*. Thus, in the first step, the table *MaxIndAlternative* has to be created by Statement *S1*:

```
S1: CREATE TABLE MaxIndAlternative
      AS SELECT *
      FROM Ind i
      WHERE Conf(i) = [max(Conf(*))];
```

The final result can then be produced by Statement *S2*:

```
S2: SELECT *
      FROM R.X t, Ind i, MaxIndAlternative mia
      WHERE Lineage(t,i)
      AND i.id = mia.id
      AND i.val = mia.val
      COMPUTES Confidences;
```

E.2. Working Implementation of Query *Q2*

Query *Q2* does not work for the same reasons as *Q1* (EXISTS-predicate and vertical subquery). The EXISTS-predicate can be realized by a join and the subquery can be realized by an auxiliary table *PersonWithNonUniqueName*. Thus, in the first step, the table *PersonWithNonUniqueName* has to be created by Statement *S3*:

```
S3: CREATE TABLE PersonWithNonUniqueName
      AS SELECT DISTINCT t1.tid
      FROM Person t1, Person t2
      WHERE t1.tid != t2.tid
      AND t1.name = t2.name;
```

The final result can then be produced by Statement *S4*:

```
S4: SELECT *
      FROM PersonWithNonUniqueName
      WHERE Conf(*) = 1.0;
```

E.3. Working Implementation of Query *Q3*

Query *Q3* does not work for the same reasons as *Q1* and *Q2* (IN-predicate and vertical subquery). The IN-predicate can be realized by a join and the subquery can be realized by an auxiliary table *LessCertainDecision*. Thus, in the first step, the table *LessCertainDecision* has to be created by Statement *S5*:

```
S5: CREATE TABLE LessCertainDecision
      AS SELECT id
      FROM Ind i
      WHERE [max(Conf(*))] < 0.6;
```

¹¹Status: December 2011.

The final result can then be produced by Statement S6:

```
S6:  SELECT      *
      FROM        R.X t, Ind i, LessCertainDecision lcd
      WHERE       Lineage(t,i)
      AND         i.id = lcd.id
      ORDER BY    Confidences ASC;
```

Note, the NOT IN-predicate and the NOT EXISTS-predicate cannot be realized by using theta-joins only, but also require a set minus operator (EXCEPT or MINUS) or an outer join (LEFT OUTER JOIN or RIGHT OUTER JOIN). Although they are listed in the TriQL manual, outer joins are not implemented in the Trio version currently available. In contrast, a set minus operator is part of this Trio version, but does not work correctly. Therefore, using queries based on one of these subquery predicates is more complicated at the moment, but will not pose a problem, when a version of the Trio prototype will be publicly accessible that supports all the query features listed in the TriQL manual.

F. EXPERIMENTAL RESULTS

In this section, we present the results of our experiments on efficiency in more detail. All these experiments were performed on machine with an Intel(R) 2.8GHz dual core processor, 4GB main memory, and a 64-bit operating system. We conducted a set of $(\alpha, 1 - \alpha)$ -restrictions with different threshold settings and the original *W-graph*-generation mapping ν . Moreover, we analyze the improvement of efficiency that result by additionally using an HC-restriction (*W-graph*-generation mapping ν_{HC}) in Section F.4. In all experiments, we use *M-graph* decomposition for making our indeterministic deduplication approach feasible in practice. The setting $\alpha = 0.0165$ was the lowest threshold for which we were able to perform an $(\alpha, 1 - \alpha)$ -restriction with the *W-graph*-generation mapping ν . In contrast, by using the mapping ν_{HC} (HC-restriction), we were able to perform an $(\alpha, 1 - \alpha)$ -restriction up to the threshold setting $\alpha = 0.002$.

F.1. Algorithm Complexity

In Table IV, we present our results on algorithm complexity for $(\alpha, 1 - \alpha)$ -restrictions with the *W-graph*-generation mapping ν (without HC-restriction). We measured complexity by the number of definite positive edges (matches, i.e., edges weighted with $\gamma = 1$), the number of uncertain edges (possible matches, i.e., edges weighted with $0 < \gamma < 1$), the number of *partial M-graphs* in which the initial *M-graph* could be decomposed, the number of *partial W-graphs* that could be derived from these *partial M-graphs*, the number of consistent *partial W-graphs*, and runtime. As mentioned in Section 7, runtime started from processing the initial *M-graph* and hence did not include tuple matching.

The results presented in Table IV show that the number of *partial W-graphs* only linearly increased with a growing number of uncertain edges, if this number remained low, but increased exponentially, if this number exceeds 50.

Moreover, in Table V, we present the absolute frequencies of the different sizes of *partial M-graphs* (frequency distribution of *partial M-graph* size) of several $(\alpha, 1 - \alpha)$ -restrictions. Note, an *M-graph* of size 1 represents a unicum. The largest *M-graph* size we measured in our experiments for restrictions with nonhierarchical clusterings with the settings $\alpha \in [0.0165, 0.5]$ was 13. In contrast, the largest *M-graph* size we measured in our experiments for restrictions with hierarchical clusterings with the settings $\alpha \in [0.002, 0.5]$ was 213. Per definition, these frequencies are independent from the use of an HC-restriction. Nevertheless, additionally using an HC-restriction

Table IV. Algorithm Complexity of Several $(\alpha, 1 - \alpha)$ -Restrictions with *M-Graph* Decomposition and the *W-Graph-Mapping* ν

α	#definite positive edges	#uncertain edges	#partial <i>M-graphs</i>	#partial <i>W-graphs</i>	#consistent partial <i>W-graphs</i>	runtime [sec.]
0	9	577004	$\rightarrow \infty^*$	$\rightarrow \infty^*$	$\rightarrow \infty^*$	$\rightarrow \infty^*$
.0165	9	98	1905	133296	3085	18.519
.0175	9	95	1907	4288	2119	10.549
.02	9	84	1915	2354	2041	8.644
.03	9	60	1935	2122	2015	4.926
.04	9	43	1951	2018	1997	2.581
.05	9	42	1952	2011	1996	2.519
.06	9	41	1953	2011	1996	2.406
.07	10	37	1955	2003	1992	2.232
.08	10	33	1958	1997	1990	2.013
.09	10	33	1958	1997	1990	1.984
.1	10	30	1961	1996	1990	1.845
.2	11	13	1977	1991	1989	1.131
.3	15	4	1982	1986	1986	1.009
.4	16	2	1983	1985	1985	0.969
.5	17	0	1984	1984	1984	0.966

*Due to our limited resources, processing a full-indeterministic approach was not feasible.

Table V. Absolute Frequencies of Different Partial *M-Graph* Sizes of Several $(\alpha, 1 - \alpha)$ -Restrictions with *M-Graph* Decomposition and the *W-Graph-Mapping* ν

α	frequency of partial <i>M-graph</i> size										
	1	2	3	4	5	6	7	8	9	...	13
.0165	1848	42	8	2	2	1	1	0	0	...	1
.0175	1850	41	8	3	2	1	1	0	1	...	0
.02	1858	42	8	4	2	0	0	1	0	...	0
.03	1886	40	6	1	1	0	1	0	0	...	0
.04	1912	31	7	0	1	0	0	0	0	...	0
.05	1913	31	7	1	0	0	0	0	0	...	0
.06	1915	30	7	1	0	0	0	0	0	...	0
.07	1918	29	8	0	0	0	0	0	0	...	0
.08	1922	30	6	0	0	0	0	0	0	...	0
.09	1922	30	6	0	0	0	0	0	0	...	0
.1	1927	29	5	0	0	0	0	0	0	...	0
.2	1955	21	1	0	0	0	0	0	0	...	0
.3	1965	16	1	0	0	0	0	0	0	...	0
.4	1967	15	1	0	0	0	0	0	0	...	0
.5	1969	14	1	0	0	0	0	0	0	...	0

allowed us to perform experiments also for extreme small values of α (i.e., $\alpha < 0.01$) and hence supply us more results on frequency distribution. However, distributions having frequencies up to the size of 213 are hard to present in a single table. For that reason, Table V only shows the results for the indeterministic approach performed with the non-hierarchical *W-graph-mapping* ν .

In large restrictions ($\alpha > 0.06$), the initial *M-graph* could be completely decomposed in partial *M-graphs* with one node (unicum), two nodes, or three nodes, respectively. In contrast, the smaller the restrictions, the less the number of *partial M-graphs*

Table VI. Data (Storage) Complexity Resulting from Several $(\alpha, 1 - \alpha)$ -Restrictions with *M-Graph* Decomposition and the *W-Graph*-Mapping ν

α	#result tuples	#indicator alternatives	#unicums
0	$\rightarrow \infty^*$	$\rightarrow \infty^*$	12
.0165	5739	1194	1848
.0175	2349	227	1850
.02	2156	140	1858
.03	2091	88	1886
.04	2039	53	1912
.05	2036	51	1913
.06	2035	50	1915
.07	2027	44	1918
.08	2022	37	1922
.09	2022	37	1922
.1	2019	33	1927
.2	2001	12	1955
.3	1990	4	1965
.4	1987	2	1967
.5	1984	0	1969

*Due to our limited resources, processing a full-indeterministic approach was not feasible.

the initial *M-graph* could be decomposed into and the larger the resultant *partial M-graphs* became.

F.2. Data Complexity

In Table VI, we present our results on the complexity of the resultant data measured by the number of resultant tuples, the number of indicator alternatives required to model all resultant complementations, and the number of unicums, that is, tuples that are nonduplicates with absolute certainty.

From $\alpha = 0.5$ to $\alpha = 0.04$, the increase of resultant tuples and the amount of required indicator alternatives remained low, with or without an HC-restriction. Moreover, the most tuples were unicums. For smaller values of α , storage complexity increased dramatically. For example, for $\alpha = 0.0165$ from 2000 input tuples 5739 output tuples and 1194 indicator alternatives resulted.

F.3. Data (Un)certainity

In Table VII, we present the (un)certainity of the resultant data measured by the number of resultant possible worlds, *Uncertainty Density*, and *Answer Decisiveness*.

The number of modeled possible world increased exponentially with a growing area of indeterministically handled decisions. However, *Uncertainty Density* as well as *Answer Decisiveness* remained close to the optimal value of complete certainty, even if α was very low. That's because the amount of unicums outweighs the amount of nonunicums by far.

F.4. Efficiency Improvement by HC-Restrictions

For demonstrating the ability of HC-restrictions to improve efficiency, we conduct the same set of experiments as before, but now with the hierarchical *W-graph*-generation mapping ν_{HC} . The results on algorithm complexity, data complexity and data (un)certainity are presented in Table VIII and Table IX. Note, per definition the

Table VII. Data (Un)certainty Resulting from Several $(\alpha, 1 - \alpha)$ -Restrictions with *M-Graph* Decomposition and the *W-Graph*-Mapping ν

α	#possible worlds	Uncertainty Density	Answer Decisiveness
0	$\rightarrow \infty^*$	$\rightarrow 1^*$	$\rightarrow 0^*$
.0165	$3.599 \cdot 10^{23}$	0.0151	0.9915
.0175	$1.045 \cdot 10^{23}$	0.0153	0.9911
.02	$1.205 \cdot 10^{21}$	0.0149	0.9915
.03	$4.696 \cdot 10^{15}$	0.0117	0.9934
.04	$2.378 \cdot 10^{11}$	0.0088	0.9939
.05	$1.698 \cdot 10^{11}$	0.0087	0.9939
.06	$8.493 \cdot 10^{10}$	0.0084	0.9940
.07	$7.644 \cdot 10^9$	0.0078	0.9942
.08	$1.019 \cdot 10^9$	0.0073	0.9947
.09	$1.019 \cdot 10^9$	0.0073	0.9947
.1	$1.698 \cdot 10^8$	0.0067	0.9951
.2	4,096	0.0030	0.9973
.3	16	0.0010	0.9989
.4	4	0.0005	0.9994
.5	1	0	1

*Due to our limited resources, processing a full-indeterministic approach was not feasible.

number of definite positive edges, the number of uncertain edges and the number of *partial M-graphs* (as well as its frequency distribution) are not affected by the hierarchical clustering and hence remained unchanged.

The presented results show that the additional HC-restriction improved efficiency enormously for small $(\alpha, 1 - \alpha)$ -restrictions, even if not in terms of runtime. That is because the decomposition of the initial *M-graph* is in general one of the most-time consuming steps, especially if the size of the resultant *partial W-graphs* is relatively small. Since the complexity of decomposition depends on the number of uncertain edges that is not affected by the HC-restriction, the runtime of an $(\alpha, 1 - \alpha)$ -restriction combined with an HC-restriction did not decrease noticeably compared to the runtime of an $(\alpha, 1 - \alpha)$ -restriction without an HC-restriction. The reason that runtime significantly grew further on for small values of α was the enormous sizes of the created *W-graphs* (up to 213).

In contrast, the number of *partial W-graphs* and the number of consistent *partial W-graphs* did not grow exponentially anymore and was always lower than 1994 graphs. An interesting observation is that the number of consistent *partial W-graphs* initially grew from 1984 to 1989, if the area of indeterministically handled decisions was increased from $\alpha = 0.5$ to $\alpha = 0.2$, but then shrank up to 1766 for $\alpha < 0.2$. In contrast, the total number of *partial W-graphs* initially increased from 1984 ($\alpha = 0.5$) to 1993 ($\alpha = 0.0175$), then varied between the values 1980 and 1991 for the settings $\alpha \in (0.005, 0.02)$, and finally shrank noticeably up to 1891 for $\alpha = 0.002$. Data complexity and data uncertainty (Table IX) stayed low even for small values of α . Moreover, instead of a dramatical increase, the number of required indicator alternatives shrank surprisingly from 69 to 45, if α was lower than 0.005.

Another difference to the nonhierarchical clustering results in the number of unicums, for example, 1848 unicums for mapping ν and 1869 unicums for mapping ν_{HC} for the setting $\alpha = 0.0165$. Whereas, by using the *W-graph*-generation mapping ν , the number of unicums is equivalent to the number of *partial M-graphs* with exactly

Table VIII. Algorithm Complexity of Several $(\alpha, 1 - \alpha)$ -Restrictions with *M-Graph* Decomposition and the *W-Graph*-Mapping ν_{HC}

α	#definite positive edges	#uncertain edges	#partial <i>M-graphs</i>	#partial <i>W-graphs</i>	#consistent partial <i>W-graphs</i>	runtime [sec.]
0	9	577004	$\rightarrow \infty^*$	$\rightarrow \infty^*$	$\rightarrow \infty^*$	$\rightarrow \infty^*$
.002	9	618	1725	1891	1766	497.867
.003	9	359	1779	1923	1828	232.122
.004	9	207	1830	1970	1883	69.568
.005	9	169	1851	1984	1912	46.942
.006	9	161	1857	1984	1918	41.839
.007	9	155	1862	1983	1921	35.488
.008	9	149	1867	1983	1926	31.518
.009	9	148	1868	1983	1927	32.467
.01	9	137	1872	1980	1928	25.104
.0165	9	98	1905	1991	1963	10.886
.0175	9	95	1907	1993	1965	11.143
.02	9	84	1915	1992	1970	8.575
.03	9	60	1935	1991	1980	4.986
.04	9	43	1951	1991	1983	2.620
.05	9	42	1952	1991	1984	2.484
.06	9	41	1953	1991	1984	2.374
.07	10	37	1955	1989	1983	2.267
.08	10	33	1958	1990	1985	1.999
.09	10	33	1958	1990	1985	1.982
.1	10	30	1961	1990	1986	1.874
.2	11	13	1977	1989	1989	1.118
.3	15	4	1982	1986	1986	1.028
.4	16	2	1983	1985	1985	0.937
.5	17	0	1984	1984	1984	0.902

*Due to our limited resources, processing a full-indeterministic approach was not feasible.

one node, by using the mapping ν_{HC} unicums can also result from *partial M-graphs* with more than one node. An example of such a situation is illustrated in Figure 23. The initial *M-graph* M_1 has one definite negative edge $\{t_1, t_2\}$ and two uncertain edges $\{t_1, t_3\}$ and $\{t_2, t_3\}$. From using the nonhierarchical clustering mapping ν none of the three tuples would be a unicum, because t_1 and t_3 would be identical in one world and t_2 and t_3 would be identical in another world. In contrast, from applying the mapping ν_{HC} only the three *W-graphs* G_1 , G_2 , and G_3 result. Only two of these *W-graphs*, namely G_1 and G_2 , are consistent. Since t_1 is not connected with any other tuple in one of these two consistent *W-graphs*, t_1 is a unicum, despite of the fact that it is involved in an *M-graph* with more than one node. In conclusion, the number of unicums resulting from an $(\alpha, 1 - \alpha)$ -restriction is always equal or greater by using an additional HC-restriction than by using without.

A further interesting aspect is that the number of resultant possible worlds was dramatically increasing (up to $3.6 \cdot 10^{23}$) for a shrinking value of α in our non-hierarchical approach (*W-graph*-generation mapping ν), but was always lower than $2 \cdot 10^{19}$ by using an HC-restriction. Moreover, by additionally using an HC-restriction, the number of possible worlds decreased for settings $\alpha < 0.005$.

To illustrate the improvement of efficiency further on, we compare the number of generated *partial W-graphs*, the number of generated consistent *partial W-graphs*, the

Table IX. Data (Storage) Complexity and Data (Un)certainty Resulting from Several $(\alpha, 1 - \alpha)$ -Restrictions with *M-Graph* Decomposition and the *W-Graph*-Mapping ν_{HC}

α	#result tuples	#indicator alternatives	#unicums	#possible worlds	Uncertainty Density	Answer Decisiveness
0	$\rightarrow \infty^*$	$\rightarrow \infty^*$	≥ 12	$\rightarrow \infty^*$	$\rightarrow 1^*$	$\rightarrow 0^*$
.002	2064	45	1901	$1.359 \cdot 10^{11}$	0.0096	0.9953
.003	2070	53	1883	$2.505 \cdot 10^{14}$	0.0123	0.9948
.004	2062	59	1875	$2.630 \cdot 10^{16}$	0.0135	0.9936
.005	2069	69	1861	$1.818 \cdot 10^{19}$	0.0156	0.9927
.006	2067	68	1863	$9.089 \cdot 10^{18}$	0.0155	0.9931
.007	2063	64	1867	$5.112 \cdot 10^{18}$	0.0154	0.9931
.008	2063	64	1867	$2.272 \cdot 10^{18}$	0.0152	0.9934
.009	2063	64	1867	$2.272 \cdot 10^{18}$	0.0151	0.9934
.01	2060	59	1872	$2.840 \cdot 10^{17}$	0.0143	0.9935
.0165	2057	65	1869	$7.213 \cdot 10^{17}$	0.0143	0.9927
.0175	2055	66	1870	$1.154 \cdot 10^{18}$	0.0144	0.9924
.02	2051	62	1875	$2.886 \cdot 10^{17}$	0.0140	0.9929
.03	2040	50	1895	$1.113 \cdot 10^{14}$	0.0114	0.9940
.04	2023	35	1921	$2.446 \cdot 10^{10}$	0.0085	0.9945
.05	2023	35	1921	$2.446 \cdot 10^{10}$	0.0085	0.9945
.06	2022	34	1923	$1.223 \cdot 10^{10}$	0.0082	0.9946
.07	2018	30	1926	$3.057 \cdot 10^9$	0.0077	0.9947
.08	2017	27	1927	$1.019 \cdot 10^9$	0.0073	0.9949
.09	2017	27	1927	$1.019 \cdot 10^9$	0.0073	0.9949
.1	2015	25	1931	$1.699 \cdot 10^8$	0.0067	0.9951
.2	2001	12	1955	4,096	0.0030	0.9973
.3	1990	4	1965	16	0.0010	0.9989
.4	1987	2	1967	4	0.0005	0.9994
.5	1984	0	1969	1	0	1

*Due to our limited resources, processing a full-indeterministic approach was not feasible.

Fig. 23. The initial *M-graph* M_1 , its three *W-graphs* G_1 to G_3 and the unicum t_1 .

number of resultant tuples, the number of resultant indicator alternatives, the number of resultant unicums, and the number of resultant possible worlds with and without using an additional HC-restriction in Figures 24–29.

All six figures show that the complexity (algorithm complexity in terms of number of *partial W-graphs* and number of consistent *partial W-graphs*, and data (storage) complexity in terms of number of resultant tuples and number of resultant indicator alternatives) as well as data uncertainty (in term of number of possible worlds) dramatically increased for $\alpha \leq 0.03$, if the non-hierarchical *W-graph*-generation mapping ν was used. In contrast, by using the hierarchical *W-graph*-generation mapping ν_{HC} the

Fig. 24. Number of *partial W-graphs* for $0.5 \geq \alpha \geq 0.002$ with/without using an HC-restriction.

Fig. 25. Number of consistent *partial W-graphs* for $0.5 \geq \alpha \geq 0.002$ with/without using an HC-restriction.

Fig. 26. Number of resultant tuples for $0.5 \geq \alpha \geq 0.002$ with/without using an HC-restriction.

results remained stable, even for small settings of α . Moreover, as already discussed above, the number of unicums resulting from an $(\alpha, 1 - \alpha)$ -restriction combined with an HC-restriction was always higher than 1980. In contrast, for small settings of α the number of unicums became less than 1700 (and theoretically only 12 for a full-indeterministic approach), if no HC-restriction was used.

Fig. 27. Number of indicator alternatives for $0.5 \geq \alpha \geq 0.002$ with/without using an HC-restriction.

Fig. 28. Number of resultant unicums for $0.5 \geq \alpha \geq 0.002$ with/without using an HC-restriction.

Fig. 29. Number of possible worlds (measured in $lb(\#possible\ worlds)$) for $0.5 \geq \alpha \geq 0.002$ with/without using an HC-restriction.

In summary, the benefits of the HC-restriction were that $(\alpha, 1 - \alpha)$ -restrictions became computable even for small values of α , because of the reduced number of *partial M-graphs*, and that the data (storage) complexity was manageable even for large areas of indeterministically handled decisions ($\alpha < 0.01$).