Distributed Transition System with Tags and Value-wise Metric, for Privacy Analysis

Siva Anantharaman[1], **Sabine Frittella**[2,*], **Benjamin Nguyen**[3]

[1] LIFO, Université d'Orléans (France), email: siva@univ-orleans.fr

[2] INSA, Centre Val de Loire (France), email: sabine.frittella@insa-cvl.fr

[3] INSA, Centre Val de Loire (France), email: benjamin.nguyen@insa-cvl.fr

Abstract. We propose in this work a logical framework to formally model how a given piece of *private* information P (specified as a set of tuples) on a given database D, can get captured by an adversary A with a sequence of queries on the database; at every stage of the querying process, the knowledge acquired on D with the answers to his/her current and earlier queries, is 'saturated' using relational deductions; for this saturation, external public data given in advance can also be used. We assume that the database D can be protected with generalization mechanisms. The logical framework will be built on concepts from Probabilistic Automata, Probabilistic Concurrent Systems, and Probabilistic labeled transition systems, nd will be named *Distributed Labeled Tagged Transition System* (DLTTS). A couple of concrete examples will show how the DLTTS can be used in practice. A second important point addressed in this work is in the realm of what is known as 'differential privacy'; its object is to study the situations where the query-answering mechanism (of D's DBMS) may or may not distinguish 'sufficiently/properly' the answers to two different instances of queries (by adversary A). A classical result in this direction says the following: If D, D' are two databases (with all data of the same type and of the same length for both) adjacent for the Hamming metric (they differ at most on a single entry), then the answers to certain queries on D, D' are not properly distinguishable. In our current work, we shall show that on a larger and more general class of databases – where the data could be of mixed types, nd not necessarily of the same length – (partial) metrics can be defined 'value-wise', i.e., in a constructive approach, and more general notions of adjacency between data bases can be defined, using these metrics; we will show them to be finer for differential privacy analysis.

Keywords: Database, Privacy, Transition System, Probability, Distribution.

1 Introduction

Data anonymization has been investigated for decades, and many privacy models have been proposed (k-anonymity, *differential privacy*, …) whose goals are to protect sensitive information. In this paper, our goal is not to define a new privacy model, but rather to propose a logical framework (agnostic to the privacy model) to formally model how the information stored in a database can get captured progressively by any agent repeatedly querying the database, i.e., we consider an agent with authorized access to the database, asking successively, *different* legitimate queries, and using the results of these queries, possibly along with additional background information, to derive information on the individuals in the database. One can envision several applications of our model.

[*] The research of Sabine Frittella was funded by the grant ANR JCJC 2019, project PRELAP (ANR-19-CE48-0006).

A first application is forensics: for instance, in the case of a dataleak, by analyzing the queries asked by each user, our model is able to quantify the knowledge each user will eventually obtain, and thus find which user has the highest probability of being responsible for the said dataleak.

A second application is to quantify and detect potential dynamic reidentification attacks on a database by a user. Such detection could lead to preventive measures such as not answering any more queries of this specific user. In this case, our approach functions as a monitor, updating its knowledge after each user query.

We start with the observation that databases are distributed over several 'worlds' in general (in a probabilistic sense), and querying such databases leads to answers which would also be distributed; and conceivably, to such distributed answers one could assign probability distributions of pertinence to the query.

Example 1 (Database worlds)**.** Consider the following database.

Name	Age	Gender	Dept.	Ailment
Alice	[29-30]	F	Chemistry	Flu

Table 1: Example database D_{ex1}

This database is distributed over the following two possible worlds (underlying databases):

Name	Age	Gender	Dept.	Ailment
Alice	29	F	Chemistry	Flu

Table 2: Possible world Ω^1_{ex1}

Name	Age	Gender	Dept.	Ailment
Alice	30	F	Chemistry	Flu

Table 3: Possible world Ω^2_{ex1}

If an observer has no other information about the possible worlds, then (s)he will assign equal probabilities of $P(\Omega^1_{ex1}|D_{ex1}) = P(\Omega^2_{ex1}|D_{ex1}) = 0.5$ of representing 'reality'. If an observer knows for instance that Alice is in her twenties, then (s)he will assign probabilities $P(\Omega^1_{ex1}|D_{ex1} \wedge \text{twenties}) = 1$ and $P(\Omega^2_{ex1}|D_{ex1} \wedge \text{twenties}) = 0$. □

The logical framework presented in this article, named DLTTS (*Distributed Labeled Tagged Transition System*), formally models how a given *private* information P, stored on a given (distributed) database D, can get captured progressively by an adversary querying the database repeatedly. To every node on a DLTTS will be attached a *tag* (that we formally define later), representing the 'current' knowledge of the adversary, acquired from the responses to his/her queries by the answering mechanism, at the nodes traversed on a run. This knowledge can be 'saturated' by the adversary at any node on the run, using (a given class of) relational deductions, in combination with public information from certain external databases, given in advance. This formalism can thus be used to better understand how a user querying a database is able to build his/her 'knowledge'.

This formal model also has an additional ingredient, a *distance*, which can be suitably defined, if necessary, for the control/analysis mechanisms for drawing certain correct conclusions regarding possible database values.

Ideas from Related Work and their Influence. As was observed in Example 1 above, databases could in general be distributed over several 'worlds'; so, querying such bases also leads in general to answers which would be distributed; to such distributed answers one could conceivably assign probability distributions of relevance to the query. It would therefore be natural to consider the probabilistic automata of Segala ([14, 15]), with outputs, as a first approximation for the logical structure we are looking for, for formally analyzing the evolution of the distributed information. The formalism of Distributed Transition Systems (DTS) appeared a little later; but almost all of these had as objective the behavioral analysis of the distributed transitions, based on traces and/or on simulation/bisimulation. Quasi- or pseudo- or hemi- metrics, suitably defined, were employed in the reasonings, for instance in [4, 5, 8]. But the pseudo- (or hemi-) metrics they use do not seem extendable suitably into a notion of (even partial) metric between data and databses, based on which one could have conceived novel notions of adjacence. On the other hand, [6] has presented an approach based on (pseudo- /hemi-) metrics, for defining novel and 'functorial' notions of ajacence between databases, which could serve to refine notions of differential privacy; but that work considers only ' 'classical' databases in standard formats (numerical or strings, all of the same dimension)/

Thus, although our lookout for a metric based vision for privacy analysis has been influenced by the formalisms developed in many of these works, but our objectives are quite different. As the developments in this work will show, our *syntax*-based metric can almost directly handle data of 'mixed types': they can be numbers or literals , but can also be 'anonymized' as intervals or sets; they can also be taxonomically related to each other on a tree structure.

As will be shown in Section 7, the partial metric that we shall build 'data-wise' (in a constructive approach') on type compatible sets of data, can be used for defining a novel notion of adjacence between databases; it has also led us to a finer notion of ϵ-distinguishabilty on mechanisms answering queries. The practical application for the DLTTS vision that we have presented in Section 8 of the current paper, is an addition to our earlier work [1]. Although rather simple, it must be sufficiently illustrative of how the DLTTS vision can be used. On the other hand, the syntactic developments presented in Section 8 have certain similarities, in our opinion, with the semantic considerations presented in [7].

Contributions. Our paper makes the following contributions:

- We propose a formal model, named DLTTS, to represent how the information stored in a database, protected via a generalization mechanism, is progressively captured by a user querying it.
- We show how to build an adequate distance function to be 'plugged' into the DLTTS, if needed for the purposes of control or analysis; and present an example using such a distance, that we call 'value-wise' metric.
- We show how to extend the use of DLTTS to databases protected via *differential privacy* mechanisms.
- We show how DLTTS could be used in practice through the analysis of an example use case. We consider a scenario where the owner of a database containing generalized information

wants to evaluate the maximal probability of information leakage. This can be seen as a form of quantitative privacy risk assessment.

Structure of the paper. Section 2 presents preliminaries, notations and our running example. Section 3 introduces DLLTS. Section 4 proposes a 'value-wise' metric. Sections 5 and 6 present the notions of ϵ-distinguishability, ϵ-local differential privacy, and ϵ-differential privacy for databases. In Section 7, we show that the notions of ϵ-distinguishability and ϵ-differential privacy can be refined by using our metric rather than the usual Hamming metric. Section 8 shows how DLTTS works in practice through a case study. Section 9 concludes the paper.

Positioning w.r.t. our earlier work. The work presented in this paper is an extensively revisited version of our earlier work [1], which was presented in a conference. In that work, the formalism of DLTTS was already developed, as well as a 'value-wise' (partial) metric between type compatible sets of data, for a large class of databases. The same role continues to be played by the DLTTS in the current paper, with some additional precisions; however *there is a major modification in the current version*, as regards the 'value-wise' metric: this metric is now assumed known *only* to (the system administrator and) the oracle mechanism $\mathcal{O}$ controlling the runs on a DLTTS modeling a query-sequence by an adversary on the base. More importantly, contrary to [1], the oracle $\mathcal{O}$ is no longer assumed to give any information of any kind to the adversary (such as e.g., on how close or how far (s)he is, from information intended to remain secret), at any node on the DLTTS. Thus our new proposition can now be used as an 'assistant' for a database administrator to control and limit query execution on a sensitive database. In particular, Section 8 is a new contribution, Section 4 has been extended, and other sections have been restructured and augmented with novel DLTTS examples and illustrations to enhance readability. Sections 5, 6 and 7 are presented similarly as in [1].

2 Preliminaries

In this section, we present concepts useful to understand the work presented in this article, and introduce a simple running example.

2.1 Useful concepts definitions

We assume given a database D, with its attributes set $\mathcal{A}$, usually divided in three disjoint groups: the subgroup $\mathcal{A}^{(i)}$ of *identifiers*, $\mathcal{A}^{(qi)}$ of *quasi-identifiers*, and $\mathcal{A}^{(s)}$ of *sensitive attributes*. The tuples of database D will be generally denoted as t, and their attributes denoted respectively as t^i, t^{qi}, and t^s in the three subgroups of $\mathcal{A}$. The attributes t^i on any tuple t of D are conveniently viewed as defining a 'user' or a 'client' stored in database D. Quasi-identifiers[1] are informally defined as a set of public attributes, which in combination with other attributes and/or external information, can allow to re-identify all or some of the users to whom the information refers.

The data in the databases will all be with finite domain, of any of the following 'basic' types: numerical, non-numerical, or literal. Some data could be structured in a complex taxonomical relation (e.g. an ontology), but we shall only consider simple tree-structured taxonomies in this work. Part

[1] The notion of quasi-identifier attributes was introduced in informal terms, by T. Dalenius in [9]. Suffices, for now, to see them as attributes that are not identifiers nor sensitive.

of the data could also be 'anonymized' via a generalization mechanism (e.g. finite intervals or finite sets, instead of precise values), over the basic types. We therefore consider the types of the data in such an extended/overloaded sense. (cf. Example 2 below)

By a *privacy policy* $P = P_A(D)$ on D with respect to a given agent/adversary A is meant the stipulation that for a certain *given set* of tuples $\{t \in P \subset D\}$, the sensitive attributes t^s on any such t shall remain inaccessible to A ('even after further deduction' – see below).

Repeatedly querying a base D is necessary, in general, to capture sensitive data meant to remain hidden under the privacy policies on D. The framework DLTTS proposed below is therefore a logical model for the evolution of the 'knowledge' that an adversary A can gain with a query-sequence on D. The base signature Σ for this framework DLTTS is assumed to be first-order, with countably many variables, finitely many constants (including dummy symbols such as '$\star$'), and no non-constant function symbols. By 'knowledge' of A we shall mean the data that A retrieves as answers to his/her successive queries, as well as other data that can be derived under relational operations on these answers, and some others derivable from these using relational combinations with data (possibly involving certain users of D) from finitely many *external databases given in advance*, denoted as $B_1, \ldots, B_m$, to which the adversary A has free access. These relational and querying operations are all assumed done with a well-delimited fragment of the relational language SQL, *assumed part of the signature* Σ. In addition, if $n \geq 1$ is the length of the data tuples in D, finitely many predicate symbols $\mathcal{K}_i, 1 \leq i \leq n$, each $\mathcal{K}_i$ of arity i, will also be part of the signature Σ; in the work presented here they will be the only predicate symbols in Σ; the role of these symbols is to allow us to see any data tuple of length $r, 1 \leq r \leq n$, as a variable-free first-order formula with top symbol $\mathcal{K}_r$, with all arguments assumed typed implicitly (with the headers of D). But, in practice, we shall drop these top symbols $\mathcal{K}_i$ and see any data tuple (not part of the given privacy policy $P_A(D)$) directly as a first-order variable-free formula over Σ; tuples t that are elements of the policy $P_A(D)$ are just written as $\neg t$ for convenience, in general. We also assume that the given external bases $B_1, \ldots, B_m$ – to which A could resort, for deducing additional information with relational operations – are of the same signature Σ as D. Thus all the knowledge A can derive from repeated queries on D can be expressed as first-order variable-free formulas over Σ.

The DLTTS framework will be shown to be well suited for capturing the ideas of acquiring knowledge and of policy violation, in an elegant and abstract setup. The definition of this framework (Section 3) considers only the case where the data, as well as the answers to the queries, do not involve any notion of 'noise' – by 'noise' we mean the perturbation of data by some *external* random mechanism. The DLTTS we consider will be modeling the lookout for the sensitive attributes of certain given users on a given base, by a single adversary, with a finite sequence of queries on D. (It is straightforward to extend the vision to model query-sequences by more than one 'non-communicating' users, seeking to capture possibly different privacy policies.) The formal definition of the DLTTS is best motivated by first presenting our 'running example' in informal style:

2.2 A Running Example

Example 2. Table 4 represents the records kept by the central Hospital of a Faculty, on recent consultations by the faculty staff of three Departments, in a University. 'Name' is an identifier attribute, 'Ailment' is sensitive, the others are QIDs; 'Ailment' is categorical with 3 branches: Heart-Disease, Cancer, and Viral-Infection; this latter in turn is categorical too, with 2 branches: Flu and Covid. By convention, such taxonomical relations are assumed known to public, (for simplicity of the example, we assume that *all* Faculty staff are on the consultation list of the Hospital.)

Name	Age	Gender	Dept.	Ailment
Joan	24	F	Chemistry	Heart-Disease
Michel	46	M	Chemistry	Cancer
Aline	23	F	Physics	Flu
Harry	53	M	Maths	Flu
John	46	M	Physics	Covid

Table 4: Hospital's 'secret' records

Line	Age	Gender	Dept.	Ailment
ℓ_1	[20 – 29]	F	Chemistry	Heart-Disease
ℓ_2	[40 – 49]	M	Chemistry	Cancer
ℓ_3	[20 – 29]	F	Physics	Viral-Infection
ℓ_4	[50 – 59]	M	Maths	Viral-Infection
ℓ_5	[40 – 49]	M	Physics	Viral-Infection

Table 5: Hospital's published records

The Hospital intends to keep 'secret' information concerning Covid infected faculty members; and the tuple $\neg(John, \star, \star, \star, Covid)$ is decided as its privacy policy. Other privacy policies are of course possible (e.g. $\neg(John, 46, M, \star, Covid)$) and would lead to other analysis formulations. Table 5 is published by the Hospital for the public, where the 'Age' attributes are anonymized as (integer) intervals [2]; and 'Ailment' is anonymized by an upward push in the taxonomy.

A certain person A, who met John at a faculty banquet, suspected John to have been infected with Covid; (s)he thus decides to consult the published record of the hospital for information.

Knowing that the 'John' (s)he met is 'male' and that Table 5 must contain some information on John's health, A has as choice lines 2, 4 and 5 (ℓ_2, ℓ_4, ℓ_5) of Table 5. A being in the lookout for a 'Covid-infected male named John', this choice is reduced to the last two tuples of the table, which are a priori indistinguishable because of anonymization (i.e. *generalization* as 'Viral-Infection'). Using external knowledge such as baby name statistics per year[3] as shown in Table 6, A can compute the number of babies named 'John' born between 1976 and 1985 (thus aged 40 to 49) to be approximately 340 000, and those born between 1966 and 1975 (thus aged 50 to 59) to be 511 000. He thus estimates the last tuple ℓ_5 to have a probability of $34/(34+51) = 0.4$ and ℓ_3 to have a probability of $51/(34+51) = 0.6$. This means A can assume, with probability 0.6 that 'John must be from the Physics Dept.', and goes to consult the following Covid-cases record 'publicly visible' at the faculty (Table 7), which confirms A's suspicion concerning John.

□

One of the objectives of this article is to define a formal model to capture this kind of reasoning. In the next section, we shall present our Distributed Labeled-Tagged Transition System (DLTTS) model, which we believe is well suited to assist in the modelling and detection of possible privacy policy breaches, and we will show in Section 8 how the model can be used by e.g. a database administrator to monitor potential privacy breaching queries.

[2] Note that here the generalized interval chosen for Age is non deterministic for extremum values.

[3] See US social security statistics at https://www.ssa.gov/OACT/babynames/index.html

Date of Birth	Population
...	...
1966	65015
1967	61603
1968	60906
1969	58737
1970	58496
1971	51508
1972	43163
1973	39169
1974	37618
1975	35075
1976	33979
1977	34199
1978	34018
1979	35412
1980	35277
1981	34872
1982	34707
1983	33158
1984	32611
1985	31499
...	...

Table 6: Demographics for name '*John*', source USA Social Security Administration

Dept.	Covid cases
Physics	M : 1 F : 0
Chemistry	M : 0 F : 0
Maths	M : 0 F : 0

Table 7: Faculty Covid-cases

3 Distributed Labeled-Tagged Transition Systems

The DLTTS framework presented in this section synthesizes ideas coming from several domains, such as the Probabilistic Automata of Segala [15], Probabilistic Concurrent Systems, and Probabilistic labelled transition systems [4, 5]. In the current paper we shall work with a limited first-order signature (as mentioned in the Introduction) denoted Σ, with countably many variables, finitely many constants, including certain additional 'dummies', no non-constant function symbols, and a finite set of propositional (predicate) symbols. Let $\mathcal{E}$ be the set of all variable-free formulas over Σ, and $\texttt{Ext}$ a given subset of $\mathcal{E}$. We assume given a decidable procedure $\mathcal{C}$ whose role is to 'saturate' any finite set G of variable-free formulas into a finite set $\overline{G}$, by adding a finite (possibly empty) set of variable-free formulas, using *relational operations* on G and $\texttt{Ext}$. This procedure $\mathcal{C}$ will be *internal* at every node on a DLTTS, it is assumed executed using a given *finite set of internal actions*. There will be an oracle $\mathcal{O}$ as mentioned earlier, to 'check' if the given privacy policy on the database is violated 'at the current node of the DLTTS'.

In the definition below, L will stand for a given (finite) set of ground (variable-free) first-order statements over Σ, its elements will be called *labels*. For any set S, $Distr(S)$ will stand for the set of all probability distributions with finite support, over the subsets of S.

Definition 1. A Distributed Labeled-Tagged Transition System (DLTTS), over a given signature Σ, is formed of:

- a finite (or denumerable) set S of states, an 'initial' state $s_0 \in S$, and a special state $\otimes \in S$ named 'Failure'.
- a finite set Act of action symbols (disjoint from Σ), with a special action $\delta \in Act$ named 'violation'.
- a (probabilistic) transition relation $\mathcal{T} \subset S \times Act \times Distr(S)$.
- A transition $\mathfrak{t} = (s, \alpha, \mathfrak{t}(s)) \in \mathcal{T}$ is said to be 'from' the state $s \in S$, and every $s' \in \mathfrak{t}(s)$ is a $\mathfrak{t}$-successor of s. The 'branch' of $\mathfrak{t}$ from s to s' is 'labeled' with a set of labels $l(s, s') \subseteq L$.
- a tag $\tau(s)$ attached to every state $s \in S$ other than $\otimes$, formed of finitely many ground first-order formulas over Σ. The tag at s_0 is $\{\top\}$, the tag at $\otimes$ is the empty set $\emptyset$.
- at every state $s \in S$ other than $\otimes$, a special action symbol $\iota = \iota_s \in Act$, *internal* at s, that 'saturates' $\tau(s)$ into a set $\overline{\tau}(s)$ using the procedure C.

The formulas in the tag $\overline{\tau}(s)$ attached to any state s will all have the same probability as assigned (by the distribution) to s. If the set $\overline{\tau}(s)$ of formulas turns out to be inconsistent, the oracle $\mathcal{O}$ will impose $(s, \delta, \otimes)$ as the only transition from s, which stands for 'violation' and 'Failure'. No outgoing transition or internal action is allowed from this halting state $\otimes$.

REMARK 1: (a) We shall assume our DLTTS to be fully probabilistic, in the following sense: Given a state s and a given distribution $E' \in Distr(S)$, there is at most one probabilistic transition from s with E' as its set of successors.

(b) We assume that 'No infinite set can get generated from a finite set' by saturation under the procedure C for deriving further knowledge, at any state s on a DLTTS. This corresponds to the assumption of *bounded inputs outputs*, as in e.g., [2, 3].

(c) The tags $\tau()$ on a DLTTS will be assumed 'coherent' in the following sense: For any state s on the DLTTS and any transition $\mathfrak{t}$ from s, and any $\mathfrak{t}$-successor s' of s, $\tau(s')$ is a finite set containing the (finite) set $\overline{\tau}(s) \cup l(s, s')$. (In all practical situations and examples considered below, we will actually have: $\tau(s') = \overline{\tau}(s) \cup l(s, s')$, except when s' is the state 'Failure' $\otimes$.) □

DLTTS and query-sequences on a database: The states of the DLTTS will stand for the various 'moments' of the querying sequence, while the tags attached to the states will stand for the knowledge A has acquired on the data of D 'thus far'. This knowledge consists partly in the answers to the queries (s)he made so far, then saturated with additional knowledge using the (finitely many) internal relational operations of the procedure C, between the answers retrieved (as tuples/subtuples in D) by A for his/her queries, and suitable tuples from the given external databases $B_1, \dots, B_m$. If the saturated knowledge of A, at a current state s on the DLTTS namely $\overline{\tau}(s)$, is not inconsistent, then the transition from s to its successor states represents the probability distribution over the possible results (resp. answers) of the next action (resp. query).

Note that we make no assumption on whether the repeated queries by A on D are treated *interactively, or non-interactively* by the DBMS. It appears that the logical framework would function exactly alike, in both cases.

Example 2(bis): Figure 1 below shows a DLTTS functioning on the Check-for-Covid problem of Example 2 above. On the anonymized database we agree to visualize the given privacy policy as: $P = \neg([40{-}50], \star, \star, Covid)$. The edges in the figure below are marked with the 'query expressions' of A, the labels on the branches are part of the answers to A's current queries:

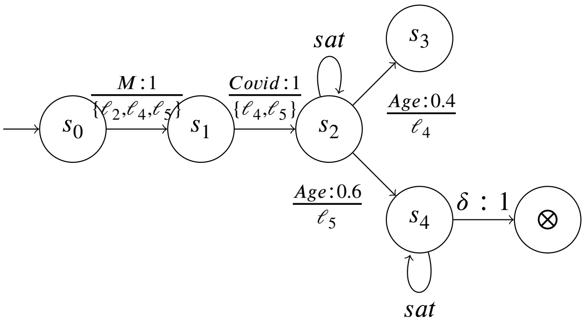


Figure 1: A DLTTS for Check-for-Covid

The set of states are: $S = \{s_0, s_1, s_2, s_3, s_4, \otimes\}$

$Act = \{M, Covid, Age, \delta\}$, where M means "Male", $Covid$ means "Covid likely",

Age means "Determine Age" and δ signifies "privacy violation".

The transitions on the DLTTS are:

- $\mathfrak{t}_0 = (s_0, Male, \{(s_1, 1)\})$. The transition $s_0 \to s_1$ is labeled with the set of tuples corresponding to males $\{\ell_2, \ell_4, \ell_5\}$.
- $\mathfrak{t}_1 = (s_1, Covid, \{(s_2, 1)\})$. The transition $s_1 \to s_2$ is labeled with the set of tuples corresponding to males for whom Covid is likely $\{\ell_4, \ell_5\}$.
- $\mathfrak{t}_2 = (s_2, Age, \{(s_3, 0.4), (s_4, 0.6)\})$. The transition $s_2 \to s_3$ (of probability 0.4) is labeled with the set of tuples corresponding to males between 50 and 59 for whom Covid is likely $\{\ell_4\}$. The transition $s_2 \to s_4$ (of probability 0.6) is labeled with the set of tuples corresponding to males between 40 and 49 for whom Covid is likely $\{\ell_5\}$.
- $\mathfrak{t}_3 = (s_6, Failure, \{(\otimes, 1)\})$.

Concerning the tags:

- There is no saturation of $\tau(s_0)$ and $\tau(s_1)$, thus each of these tags will contain a single formula indicating the possible lines of the database, i.e., $\tau(s_0) = \{\ell_0 \vee \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \vee \ell_5\}$ and $\tau(s_1) = \{\ell_2 \vee \ell_4 \vee \ell_5\}$.
- $\tau(s_2)$ is saturated by adding the (external) knowledge of Table 6, i.e. $\overline{\tau}(s_2) = \{(\ell_4 \vee \ell_5) \wedge (count(x) | x.name = John \wedge\ x.dob = 1966) = 65015 \wedge \ldots\}$
- $\tau(s_4)$ is saturated by adding the (external) knowledge of Table 7, i.e. $\overline{\tau}(s_4) = \{\ell_5 \wedge\ (x.dept = Physics \wedge x.sex = M) \to x.hasCovid = true\}$ which leads to a violation of the policy P, and explains the final transition to the $\otimes$ state.

Proposition 1. Suppose given a database D, a finite sequence of repeated queries on D by an adversary A, and a first-order relational formula $P = P_A(D)$ over the signature Σ of D, expressing the privacy policy of D with respect to A. Let $\mathcal{W}$ be the DLTTS modeling the various queries of A on D, and the evolution of the knowledge of A on the data of D, resulting from these queries and the internal actions at the states of $\mathcal{W}$, as described above.

(i) The given privacy policy $P_A(D)$ on D is violated under some run on $\mathcal{W}$, if and only if the failure state $\otimes$ on $\mathcal{W}$ is reachable on $\mathcal{W}$ under that run.

(ii) The satisfiability of the set of formulas $\overline{\tau}(s) \cup \{\neg P\}$ is decidable, at any state s on the DLTTS, under the assumptions of Remark 1(b, c).

Proof: Assertion (i) is just restatement. Observe now, that at any state s on $\mathcal{W}$, the tags $\tau(s)$, $\overline{\tau}(s)$ are both finite sets of first-order *variable-free formulas* over Σ, without non-constant function symbols. Indeed, to start with, the knowledge of A consists of the responses received for his/her queries, in the form of a finite set of data tuples/subtuples from the given bases; and by our assumption of Remark 1(b,c), no infinite set can be generated by saturating this initial knowledge with the procedure C. Assertion (ii) follows then from the known result that the inconsistency of any given finite set of *variable-free* first-order Datalog formulas is decidable, e.g., by the analytic tableaux procedure. Only the absence of variables is essential. □

4 A Metric-based vision for our Databases

Preliminary Remarks: In the analysis of the responses to query-sequences by an intruder A on databases, one has to consider the two essential issues below:

- On two different bases D, D', when are the responses to a given query by A the same (or "are indistinguishable")?

- On a given base D, when can the responses to two different query instances by A be considered "approximately the same"?

The former of the two issues above is classically addressed with a notion "adjacency" of the bases D, D': if the data in the databases are all of the same type (numerical or strings) and of the same length, then D and D' are considered (classically) adjacent iff they differ on at most one entry, i.e., iff they are at distance ≤ 1 for the so-called Hamming metric. For the latter issue, the notion of 'approximation' is not based on the Hamming metric (even on 'classical' bases), but rather on the Euclidean distance between data, and on appropriately defined probability measures.

We shall get back to these notions in some detail in Section 7. Our current objective in this section is to propose a metric-based vision that will be suitable for both the issues of analysis. We shall do that by building a particular distance function, as a partial metric between compatible sets of data tuples, on our databases of mixed types (as described in the earlier sections of the paper). It may not be the only metric-based vision that would be suitable for both the above issues; but, as will be shown in Section 7, it can already lead to a new and more general notion of adjacency between our databases, and serve as a finer tool of analysis (for the two issues above); it can also be parametrized in different ways as mentioned in Section 6.

4.1 Building a value-wise Metric for our Bases

We assume given a (distributed) database D, with the datatypes as mentioned in the Introduction (with a privacy policy P specified on D). Our objective in this section is first to look for a 'quantitative measure' for comparing type comparable tuples in $\boldsymbol{D}$, based on which it would be possible to define a (partial) notion of distance/metric – that we shall denote by ρ, between sets of tuples/subtuples in $\boldsymbol{D}$. Our objective is motivated by the following considerations. Suppose an adversary A launches a sequence of queries on $\boldsymbol{D}$, with a view to capture some of the sensitive data in the policy $\boldsymbol{P}$; and suppose a DLTTS models the sequence of queries-and-answers (as described in Section 3). Assumed given $\epsilon \geq 0$, 'sufficiently small' as a 'threshold' for approximation, the manner in which the DLTTS functions can then be refined in two different ways that we describe below:

(1) The role of the oracle mechanism $\mathcal{O}$ can be made 'sharper' than just registering the violation of the policy $\boldsymbol{P}$: if the current (saturated) knowledge of A at some node s on the DLLTS is at ρ-distance $\leq \epsilon$ to data intended secret, then the oracle $\mathcal{O}$ could force the transition from s to the state $\otimes$ Failure; such a transition would then be named *ϵ-violation* of privacy. In operational terms, a DLTTS could inform the database administrator, or some access control system, which in turn could take actions such as refusing to answer subsequent queries ;

(2) Two distinct 'query instances' by A, at some given node s on the DLTTS, could receive the same output(answer) from the answering mechanism, as concerns the secret data from $\boldsymbol{P}$. The two instances will then be said to be $epsilon$-indistinguishable, in a sense and under certain conditions that we define formally in Section 5. In such a case, the labels on the outgoing branches at s, corresponding to the knowledge gained by A from these queries, will be considered *ϵ-equivalent* in $\overline{\tau}(s)$ for privacy.

It will be actually shown in Section 5, that the notion of ϵ-equivalence can be rendered finer, into an *ϵ_ρ-equivalence*, by combining it in a suitable sense with the value-wise metric ρ that we proceed to construct now.

Remember that the knowledge of A, at any node on the DLTTS, is represented as a set of tuples, and that the data forming any tuple are assumed 'implicitly typed with the headers' of the database $\boldsymbol{D}$. For 'quantitatively' comparing two tuples of the same length, we shall assume there is a natural, injective, *type-preserving* map from one of them onto the other; this map will remain implicit in general, and two such tuples will be said to be *type-compatible*. If the two tuples are not of the same length, one of them will be projected onto (or restricted to) a suitable subtuple, so as to be type-compatible and comparable with the other; if this turns out to be impossible, the two tuples will be said to be uncomparable.

We shall propose a comparison method based on an appropriately defined notion of 'distance' between two *sets of type-compatible tuples*. For that, we shall first define a 'distance' between any two type-compatible tuples; with that purpose, we shall start with defining a notion of distance between any two data values under every given header of $\boldsymbol{D}$. As a first step, we shall therefore begin by defining, for every given header of $\boldsymbol{D}$, a binary 'distance' function on the *set of all values that get assigned to the attributes under that header*, all along the sequence of A's queries. This distance function to be defined will be a (partial) *metric*: non-negative, symmetric, and satisfying the so-called Triangle Inequality (cf. below). The '*direct-sum*' of these metrics, taken over all the headers of $\boldsymbol{D}$, will then define a (partial) metric d on the set of all type-compatible tuples of data assigned to the various attributes, under all the headers of $\boldsymbol{D}$, all along the sequence of A's queries. The 'distance' $d(t, t')$, from any given tuple t in this set to another type-compatible tuple t', will be defined as the value of this direct-sum metric on the pair of tuples (t, t'); it will be calculated 'column-wise' by definition, on $\boldsymbol{D}$ and also on the intermediary databases along A's query sequence. Note that it will a priori give us an m-tuple of numbers, where m is the number of headers (number of columns) in the database $\boldsymbol{D}$.

A single number can then be derived as the sum of the entries in the m-tuple $d(t, t')$. This sum will be denoted as $\overline{d}(t, t')$, and defined as the distance from the tuple t to the tuple t' in the database D. Finally, if S, S' are any two given finite sets of type-compatible tuples of data that get assigned to the various attributes (along the sequence of A's queries), we define the distance from the set S to the set S' as the number $\rho(S, S') = min\{\ \overline{d}(t, t') \mid t \in S,\ t' \in S'\ \}$

For clarity of presentation, in order to define the 'distance' between the data values under every given header of D, we now divide the headers of D into four classes, as below:

. 'Nominal': identities, names, attributes receiving *literal* data *not in any taxonomy* (e.g., gender, city, ...), and finite sets of such data;

. 'Numerval' : attributes receiving *numerical* values, or bounded intervals of (finitely many) numerical values;

. 'Numerical': attributes receiving *single numerical values* (numbers).

. 'Taxoral': attributes receiving *literal data in a taxonomy relation.*

• For defining the 'distance' between any two values v, v' assigned to an attribute under a given 'Nominal' header of D, for the sake of uniformity we agree to consider every value as a *finite set* of singleton values; in particular, a singleton value 'x' will be seen as the set $\{x\}$. Given two such values v, v', note first that the so-called *Jaccard Index* between them is the number $jacc(v, v') = |(v \cap v')/(v \cup v')|$, often called a 'measure of their similarity'; but this index is not a metric, because the *triangle inequality* is not satisfied; however, the Jaccard metric $d_{Nom}(v, v') = 1 - jacc(v, v') = |(v \Delta v')/(v \cup v')|$ does satisfy that property, and will suite our purposes. Thus defined, $d_{Nom}(v, v')$ is a 'measure of the dissimilarity' between the sets v and v'.

• Let $\mathcal{T}_{Nom}$ be the set of all data assigned to the attributes under the 'Nominal' headers of D, along A's queries sequence. Then the above defined binary function d_{Nom} extends to a metric on the set of all type-compatible data-tuples from $\mathcal{T}_{Nom}$, defined as the 'direct-sum' taken over the 'Nominal' headers of D.

• If $\mathcal{T}_{Num}$ is the set of all data assigned to the attributes under the 'Numerval' headers along the sequence of queries by A, we define in a similar manner (as above) a 'distance' metric d_{Num} on the set of all type-compatible data-tuples from $\mathcal{T}_{Num}$: we first define d_{Num} on any couple of values u, v assigned to the attributes under a given 'Numerval' header of D, then extend it to the set of all type-compatible data-tuples from $\mathcal{T}_{Num}$ (as the direct-sum taken over the 'Numerval' headers of D). The case where both u, v are singleton numbers, the 'distance' between them is defined down below, see 'Numerical'. We therefore assume that at least one of the two u, v is *not a singleton number*. We can then proceed exactly as previously, under the 'Nominal' headers: we agree to visualize any finite interval value as a particular way of presenting a set of numerical values (integers, usually). If one among the two values u, v is a singleton number, say a, we shall agree to see that as the interval value $[a]$. Thus defined the (Jaccard) metric $d_{Nom}([a, b], [c, d])$ will be a measure of 'dissimilarity' between $[a, b]$ and $[c, d]$.

• Between numerical data x, x' under the 'Numerical' headers, the distance we shall work with is the usual euclidean metric $|x - x'|$, normalized as: $d_{eucl}(x, x') = |x - x'|/D$, where $D > 0$ is a fixed finite number, bigger than the maximal euclidean distance between the numerical data on the databases and on the answers to A's queries.

• For the data under the 'Taxoral' headers, we choose as distance function the metric d_{wp} that we define in the Appendix (Section 10), based on the well-known notion of Wu-Palmer symmetry between the nodes of a Taxonomy tree.

REMARK 2: (a) Note that the distance computed between any couple of data, by the partial metric ρ we defined above is always a real in the interval $[0, 1]$, a 'detail' which is also true for the WP-metric between taxonomy trees, as defined in the Appendix. This plays a key role in the proof of Proposition 2, Section 7.

(b) The Hamming metric between datatuples/databases is generally well-defined only on databases with all data of a single (numerical or string) type, and all tuples of the same length. However, in the current paper, we shall be using a generalized notion of that metric, by extending that usual notion, in a natural and 'value-wise' and 'column-wise' manner as a partial metric, just as we did for the distance function ρ above. We shall denote this extension as d_h. Thus: $d_h([1,2], a), ([2,3], a)) = 1, d_h([1,2], a), ([2,3], b)) = 2$, whereas $d_h((bd, a), ([2,3], b))$ is undefined, etc.

5 ϵ-local-differential privacy, ϵ-indistinguishability

In this section we extend the result of Proposition 1 to cases where the violation of a policy can be up to a 'threshold of approximation' $\epsilon \geq 0$, in the sense we evoked in the previous section. We stick to the same notation as above. The set $\mathcal{E}$ of all variable-free formulas over Σ is thus a disjoint union of subsets of the form $\mathcal{E} = \cup\{\mathcal{E}_i^{\mathcal{K}} \mid 0 < i \leq n, \mathcal{K} \in \Sigma\}$, the index i in $\mathcal{E}_i^{\mathcal{K}}$ standing for the common length of the formulas in the subset, and $\mathcal{K}$ for the common root symbol of its formulas; each set $\mathcal{E}_i^{\mathcal{K}}$ will be seen as a database of i-tuples.

As above, we consider the situation where the queries of an adversary intend to capture certain given (sensitive) values in the database D. The following definitions of ϵ-indistinguishability (and of ϵ-distinguishability) of two different query instances for the answering mechanism $\mathcal{M}$, as well as that of ϵ-DP that will be defined in the next subsection, are essentially reformulations of the same (or similar) notions defined in [10, 11].

Definition 2. (i) Suppose the probabilistic mechanism $\mathcal{M}$, answering A's queries on the base D outputs(answers with) the same tuple $\alpha \in \mathcal{E}$ for two different input instances v, v'. Given $\epsilon \geq 0$, the two instances will be said to be ϵ-indistinguishable wrt α, if and only if:

$$Prob[\mathcal{M}(v) = \alpha] \leq e^{\epsilon} Prob[\mathcal{M}(v') = \alpha] \text{ and}$$
$$Prob[\mathcal{M}(v') = \alpha] \leq e^{\epsilon} Prob[\mathcal{M}(v) = \alpha].$$

Otherwise, the two instances v, v' are said to be ϵ-distinguishable for output α.

(ii) The probabilistic answering mechanism $\mathcal{M}$ is said to satisfy *ϵ-local differential privacy (ϵ-LDP)* for $\epsilon \geq 0$, if and only if: for any two instances v, v' of $\mathcal{M}$ *that lead to the same output*, and any set $\mathcal{S} \subset Range(\mathcal{M})$, we have

$$Prob[\mathcal{M}(v) \in \mathcal{S}] \leq e^{\epsilon} Prob[\mathcal{M}(v') \in \mathcal{S}].$$

The two small examples below illustrate ϵ-Indistinguishability:

(i) The two queries based on sub-tuples ([50–60], M, Maths) and ([40–50], M, Physics), from the Hospital's published record in Example 2 (Table 5) have both Viral–Infection as output, with respective probabilities $1/3, 2/3$; thus they are ϵ-indistinguishable for any $\epsilon \geq ln(2)$; and ϵ-distinguishable for any $0 \leq \epsilon < ln(2)$.

(ii) The 'Randomized Response' mechanism RR ([16]) can be modelled as follows. Input is (X, F_1, F_2) where X is a Boolean, and F_1, F_2 are flips of a coin (H or T). RR outputs X if $F_1 = H$, $True$ if $F_1 = T$ and $F_2 = H$, and $False$ if $F_1 = T$ and $F_2 = T$. This mechanism is $ln(3)$-LDP : the instances

$(True, H, H)$, $(True, H, T)$, $(True, T, H)$ and $(True, T, T)$ are $ln(3)$-indistinguishable for output $True$. $(False, H, H)$, $(False, H, T)$, $(False, T, H)$ and $(False, T, T)$ are $ln(3)$-indistinguishable for output $False$.

6 ϵ-Differential Privacy

The notion of *ϵ-indistinguishability of two given databases* D, D' for an adversary, is more general than that of ϵ-indistinguishability of pairs of query instances giving the same output (defined above). ϵ-indistinguishability for pairs of databases D, D' is usually defined only for bases that are *adjacent* in a certain sense (cf. below).

There seems to be no uniquely defined notion of adjacence on pairs of databases; in fact, several are known and in use in the literature. Actually, a notion of adjacence can be defined in a generic parametrizable manner (as in e.g., [6]), as follows. Assume given a map $\mathbf{f}$ from the set $\mathcal{D}$ of all databases of m-tuples (for some given $m > 0$), into some given metric space (X, d_X). The (symmetric) binary relation on pairs of databases in $\mathcal{D}$, defined by $\mathbf{f}_{adj}(D, D') = d_X(\mathbf{f}(D), \mathbf{f}(D'))$ is then said to give a *measure of adjacence* between these bases. The relation $\mathbf{f}_{adj}$ is said to define an 'adjacency relation'.

Definition 3. Let $\mathbf{f}_{adj}$ be a given adjacency relation on a set $\mathcal{D}$ of databases, and $\mathcal{M}$ a probabilistic answering mechanism for queries on the bases in $\mathcal{D}$. Two bases $D, D' \in \mathcal{D}$ are said to be $\mathbf{f}_{adj}$-indistinguishable under $\mathcal{M}$, if and only if, for any possible output $\mathcal{S} \subset Range(\mathcal{M})$, we have

$$Prob[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\mathbf{f}_{adj}(D,D')} Prob[\mathcal{M}(D') \in \mathcal{S}]$$
$$Prob[\mathcal{M}(D') \in \mathcal{S}] \leq e^{\mathbf{f}_{adj}(D,D')} Prob[\mathcal{M}(D \in \mathcal{S}].$$

The mechanism $\mathcal{M}$ is said to satisfy $\mathbf{f}_{adj}$*-differential privacy* ($\mathbf{f}_{adj}$-DP), if and only if the above conditions are satisfied for *every pair of databases* D, D' in $\mathcal{D}$, and any possible output $\mathcal{S} \subset Range(\mathcal{M})$.

Comments: (i) Given $\epsilon \geq 0$, 'classically' the notions of *ϵ-indistinguishability and of ϵ-DP* are meaningful for the choice of adjacency $\mathbf{f}_{adj} = \epsilon d_h$, where d_h is the classical Hamming metric on databases, namely, the number of 'records' where D and D' differ) and the assumption $d_h(D, D') \leq 1$, (cf. [6], the bases being classically assumed to be of the same length, all data of the same type (numerical, or string). But remember that in the current paper our databases are assumed more general, and with data of mixed types, and we shall be using an extension of the classical Hamming metric, as a partial metric between such databases.

(ii) In Section 7, we shall propose a more general notion of adjacency, based on the value-wise metric ρ that we defined in Section 4, on our databases.

(iii) On disjoint databases, it is possible to work with different adjacency relations, using different maps to the same (or different) metric space(s),

(iv) The mechanism RR described above is actually $ln(3)$-DP, not only $ln(3)$-LDP. To check DP, we have to check all possible pairs of numbers of the form $(Prob[\mathcal{M}(x) = y], Prob[\mathcal{M}(x') = y])$, $(Prob[\mathcal{M}(x) = y'], Prob[\mathcal{M}(x') = y])$, $(Prob[\mathcal{M}(x) = y], Prob[\mathcal{M}(x') = y'])$, etc., where the x, x'.... are the input instances for RR, and $y, y', ...$ the outputs. The mechanism RR has 2^3 possible input instances for (X, F_1, F_2) and two outputs (*True, False*); thus 16 pairs of numbers, the distinct ones being $(1/4, 1/4), (1/4, 3/4), (3/4, 1/4)$, $(3/4, 3/4)$; if (a, b) is any such pair, obviously $a \leq e^{ln(3)} b$. Thus RR is indeed $ln(3)$-DP. □

7 The metric ρ for Indistinguishability and DP

As was observed in the previous section, a (partial) notion of ϵ-adjacency between the databases considered in the current work can be defined using the generalized Hamming metric, by setting $\mathbf{f}_{adj} = \epsilon d_h$. Our objective now is to propose a more general notion of adjacency on our databases, based on the (partial) metric ρ defined in Section 4, and we shall show that it is a finer tool for ϵ-LDP and ϵ-DP analysis.

So, let $\mathcal{D}$ now be the set of all databases, *not necessarily all with the same number of columns, and with data of several possible types* as mentioned in the Introduction, and $\mathcal{M}$ a probabilistic answering mechanism for the queries on the bases in $\mathcal{D}$. We define then a (partial) binary relation $\mathbf{f}^{\rho}_{adj}(D, D')$ between the databases D, D' in $\mathcal{D}$ by setting $\mathbf{f}^{\rho}_{adj}(D, D') = \rho(D, D')$, visualizing D, D' as sets of type-compatible data tuples.

Given $\mathcal{M}$ and an $\epsilon \geq 0$, recall that the ϵ-indistinguishability of any two given databases for $\mathcal{M}$, and the notion of ϵ-DP for $\mathcal{M}$, were both defined in Definition 2 (Section 6); based first on a hypothetical map $\mathbf{f}$ from the set of all the databases concerned, into some given metric space (X, d_X), and an 'adjacency relation' on databases, defined as $\mathbf{f}_{adj}(D, D') = d_X(\mathbf{f}D, \mathbf{f}D')$. We now define the notion of ϵ_ρ-indistinguishability of two databases $D, D' \in \mathcal{D}$ for the mechanism $\mathcal{M}$, as well as that of ϵ_ρ-DP for $\mathcal{M}$, exactly as in Definition 2, by replacing $\mathbf{f}_{adj}$ first with the relation $\mathbf{f}^{\rho}_{adj}$, and subsequently with $\epsilon\rho$. The notions thus defined are *more general*, and are finer tools of analysis, than those presented earlier with the choice $\mathbf{f}_{adj} = \epsilon d_h$. The following example will illustrate this point.

Example 3. We go back to the 'Hospital's public record' of our previous Example 2, and the two sub-tuples ([50–60], M, Maths) and ([40–50], M, Physics), from the Hospital's published record in Example 2 (Table 5). The mechanism $\mathcal{M}$ answering two queries for 'Virus-Ailment information involving men', returns the tuples l_4, l_5 with the probability distribution $1/3, 2/3$, respectively. Let us look for the minimum value of $\epsilon \geq 0$, for which these tuples will be ϵ_ρ-indistinguishable under the mechanism $\mathcal{M}$.

We first compute the ρ-distance between the two tuples:

$$\rho(l_4, l_5) = \overline{d}(l_4, l_5) = (1 - \frac{1}{20}) + 0 + 1 + 0 = 39/20.$$

The condition for l_4 and l_5 to be ϵ_ρ-indistinguishable under $\mathcal{M}$ is thus:

$$(1/3) \leq e^{(39/20)\epsilon} * (2/3), \quad (2/3) \leq e^{(39/20)\epsilon} * (1/3).$$

Which gives: $\epsilon \geq (20/39) * ln(2)$. That is, for $\epsilon \geq (20/39) * ln(2)$, the two tuples l_4 and l_5 will be ϵ_ρ-indistinguishable; and for values of ϵ with $0 \leq \epsilon < (20/39) * ln(2)$, these tuples will be ϵ_ρ-distinguishable.

Now, the Hamming metric is definable between these two tuples: they differ only at two places, we have $d_h(l_4, l_5) = 2$. So, they are ϵ_h-indistinguishabilty (wrt d_h), for $\epsilon \geq 0$, if and only if: $(2/3) \leq e^{2\epsilon} * (1/3)$, i.e., $\epsilon \geq (1/2) * ln(2)$.

In other words, if these two tuples are ϵ_ρ-indistinguishable wrt ρ under $\mathcal{M}$ for some ϵ, then they will be ϵ_h-indistinguishable wrt d_h for the same ϵ. But the converse is not true, since $(1/2) * ln(2) < (20/39) * ln(2)$. In other words: $\mathcal{M}$ ϵ-distinguishes more finely combined with ρ, than with d_h. □

Remark 3: The statement "$\mathcal{M}$ ϵ-distinguishes more finely with ρ, than with d_h", is *always true* – not just in Example 3 – For the following reasons: the records that differ 'at some given position' on two bases D, D' are always at distance 1 for the Hamming metric d_h, by definition, whatever be the type of data stored at that position. Now, if the data stored at that position 'happened to be' numerical, the usual euclidean distance between the two data could have been (much) bigger than their Hamming distance 1; precisely to avoid such a situation, our definition of the metric d_{eucl} on numerical data 'normalized' the euclidean distance, to ensure that their d_{eucl}-distance will not exceed their Hamming distance. Thus, all the 'record/value-wise' metrics we defined above have their values in $[0, 1]$, as we mentioned earlier; so, whatever the type of data at corresponding positions on any two bases D, D', the ρ-distance between the records will never exceed their Hamming distance. That suffices to prove our statement above. The Proposition below formulates all this, more precisely:

Proposition 2. Let $\mathcal{D}_m$ be the set of all databases with the same number m of columns, over a finite set of given data, and $\mathcal{M}$ a probabilistic mechanism answering queries on the bases in $\mathcal{D}$. Let ρ be the metric (defined above) and d_h the Hamming metric, between the databases in $\mathcal{D}$, and suppose given an $\epsilon \geq 0$.

- If two databases $D, D' \in \mathcal{D}_m$ are ϵ_ρ-indistinguishable under $\mathcal{M}$ wrt ρ, then they are also ϵ-indistinguishable under $\mathcal{M}$ wrt d_h.

- If the mechanism $\mathcal{M}$ is ϵ_ρ-DP on the bases in $\mathcal{D}_m$, then it is also ϵ-DP (wrt d_h) on these bases.

Proof: We reason exactly as in the Example 3 (and the Remark) above: For the 'record/value-wise' metric ρ defined above, the ρ-distance between the records at any pair of corresponding positions on the two bases D, D', is always in the interval $[0, 1]$, and it never exceeds their Hamming distance. □

The idea of 'normalizing' the Hamming metric between numerical databases (with the same number of columns) has already been suggested in several works (cf. e.g., [6]) for the same reasons. When only numerical databases are considered, the metric ρ we have defined above is the same as the 'normalized Hamming metric' of [6]. Thus, our metric ρ is in a sense a generalization of that notion, for handling directly bases with more general types of data such as anonymized, taxonomies, etc.

8 Using DLTTS in practise: privacy analysis

Context and motivation. In order to demonstrate how DTLLS can be used in practice, we consider a scenario where the owner of a database containing generalized information wants to evaluate the maximal probability of information leakage. This can be seen as a form of quantitative privacy risk assessment.

We thus consider a database D which has its qid columns anonymized, with specified probability thresholds on the entries in D (i.e. the database owner does not want certain tuples to be deduced with a probability greater than a certain threshold). The various adversaries (A, B, and C) considered in this privacy anaylsis have access to only some of the qids, and different kinds of background information. The objective of the analysis is to estimate the maximal probability threshold for these attackers to get access to the sensitive values of (some of) the entries in D.

We present next this example application in detail, by making use of the DLTTS approach.

8.1 Example database

Example: An enterprise **E** stores a database D, containing a sensitive value as an integer between 1 and 10, standing for the (anonymized) responses of its employees to a questionnaire on their working conditions. The qid–attributes of D are *Id, Sex, Age*, the anonymized sensitive value is 'Response'. *Age* and *Id* may be anonymized when D is rendered public.

id	Sex	Age	Response
ℓ_1	F	[30, 40]	1
ℓ_2	F	[30, 40]	8
ℓ_3	M	[30, 40]	3
ℓ_4	M	[40, 50]	7

Objective: *An estimation for the maximal probability, for different attackers to infer the response of one or more randomly chosen employees of* **E**, *with runs on suitably constructed DLTTS.*

With such a purpose, the administrator of D conceives three different 'test attacks', where the initial knowledges/beliefs of the attackers on the qid's are specified, so as to be 'sufficiently complementary'. Based on these tests, an empirical strategy will be formulated in subsection 8.3, with some parametric conditions for accepting/refusing a given query for accessing any *given* response. It is assumed that all the employees in **E** have responded to the questionnaire; none of the attackers A, B, C is assumed to have any a priori knowledge of the 'Response' of any particular employee.

8.2 Privacy attack analysis

We detail next how DLTTS can be used to analyze privacy attack: we present several realistic attackers, and compare them.

Evaluating the success of an attack: There are two types of attackers : Attackers that have some knowledge of the individual in the database for whom they wish to retrieve information (Attackers A and B below), and Attackers with no knowledge other than the database distribution (Attacker C). The efficiency of an attack is evaluated by first computing (using the DLTTS) the probability that an attacker interested in re-identifying a given individual will have, using no external knowledge. This gives a baseline probability of success. We then compute (using a different DLTTS) the maximal probability with which an 'informed' attacker retrieves this knowledge. If such an attack has a 'better' probability of retrieving the correct knowledge, then that attack is considered as a success. (This notion will be formalized in Section 8.3.)

Attacker A works with the following initial knowledge(belief):

Sex = F : 80%,	Age = [30-40] : 70%
Sex = M : 20%	Age = [40-50] : 30%

• A aims to capture 'preferably' the response of a female employee, age 30 to 40 years.

Attacker B works with the following initial knowledge(belief):

Sex = F : 20%	Age = [30-40] : 75%
Sex = M : 80%	Age = [40-50] : 25%

• B aims to capture 'preferably' the response of a male employee, age 30 to 40 years.

Attacker C works with Initial qid knowledge inferred from the base :

Sex = F : 50%	Age = [30-40] : 75%
Sex = M : 50%	Age = [40-50] : 25%

• C to capture the response of 'any' employee of 'any' sex, and of 'any' age. C has no knowledge beyond the distribution of the qid values in the database. In particular, C has no knowledge about possible correlations between the QIDs.

(We shall agree to rename attacker C the "*Basic Analyser*").

Preliminary Remarks for attack analysis For a DLTTS-analysis on privacy policies with probability thresholds as intended above, it is natural (even necessary) to 'compare' the probability distributions available prior to choosing a particular transition. Given an outgoing transition τ at any given node s on a DLTTS, its distribution is a *multiset* of the probability measures on the branches of τ that we denote as $\mathcal{M}_\tau$. Since our DLTTS are assumed to be 'fully probabilistic', the transition τ is uniquely determined by $\mathcal{M}_\tau$ and the set $\tau(s)$ of successor states to s (and the labels on the branches from s to its successors). On the other hand, on the DLTTS that we construct below for this analysis, the following will be assumed: at any state s on the DLTTS, and any successor state s' on any transition, the tag $\tau(s')$ at s' will be implicitly set to be the set $\tau(s) \cup l(s, s')$, where $l(s, s') \subset L$ is the label on the transition's branch from s to s'; no external information and no saturation step at any of the nodes, therefore the tags at the nodes will not be mentioned explicitly.

Now, the multisets of probability distributions (for the transitions available at any given node on a DLTTS) are totally ordered by the multiset extension $\succ$ of the natural order $>$ on numbers (reals or integers). If τ, τ' are two outgoing transitions from a state s on a DLTTS, we shall *define* τ *to have priority over* τ' if and only if $\mathcal{M}_\tau \succ \mathcal{M}_{\tau'}$. Two different outgoing transitions from a state s can have the same multiset distribution, but with different successor states; if so, neither has priority over the other, they will not be '$\succeq$-distinguishable'.

For computing the maximal probability threshold for any attack to capture sensitive data, it is thus necessary to choose, at any given node on a DLTTS, an outgoing transition τ such that $\mathcal{M}_\tau$ is maximal for the ordering $\succ$. This will be the case in the attacks presented below.

On any DLTTS $\mathcal{A}$ constructed below, and any state s on $\mathcal{A}$ where the *incoming transition has a singleton label set* of the form $\{\ell\}$, a special outgoing transition, referred to as $response(\ell)$, is assumed available; its objective is to give access to the 'response' for the entry ℓ in base D. This special outgoing transition $response()$ is in fact a switch, 'turned ON, by default'. (But it can be turned OFF, at certain states by the oracle mechanism $\mathcal{O}$ of the DLTTS, on 'considerations of strategy for secrecy' – that we shall present/discuss in section 8.3 below. Pending that discussion, the switch response() is assumed ON by default; it will be an outgoing transition with probability 1.)

Basic Analyser C: We first consider the case of Basic Analyser C, with initial knowledge inferred from the base D, and its QIDs. The DLTTS-C (Figure 2) shows how C gets access to the responses of all employees, and respective probability thresholds. The DLTTS construction is based only on the database D; the probabilities on its various branches, denoted as P_{db}, are all inferred from D.

Starting from the initial state s_0 on DLTTS-C, the probability for access to the response of employee ℓ_1 is the product of the probabilities along the branches traversed by the run, namely: $(3/4) * (1/2) * (1/2) = (3/16)$,which is also same for access to the response for entry ℓ_2; access to the responses for ℓ_3 and ℓ_4 are also the same. The transitions of DLTTS-C are of $\succ$-maximal priority at all the nodes. The maximal probability threshold for access to *any of the four responses*, as reported by Basic Analyser C, turns out to be $3/16$.

Attacker B: With DLTTS-B (Figure 3), Attacker B gets access to the responses of employees, preference to males of age 30 to 40. (P_b = probabilities from B's assigned objective.)

Maximal probability thresholds computed by B: Males $6/10$, Females $1/10$. Note: The threshold $6/10$ for males is higher than the $3/16$ reported by C.

The actual probabilities for all possible responses are: $Pr(response = 3|M) = 6/10$,
$Pr(response = 7|M) = 1/5$, $Pr(response = 1|F) = 1/10$ and $Pr(response = 8|F) = 1/10$.

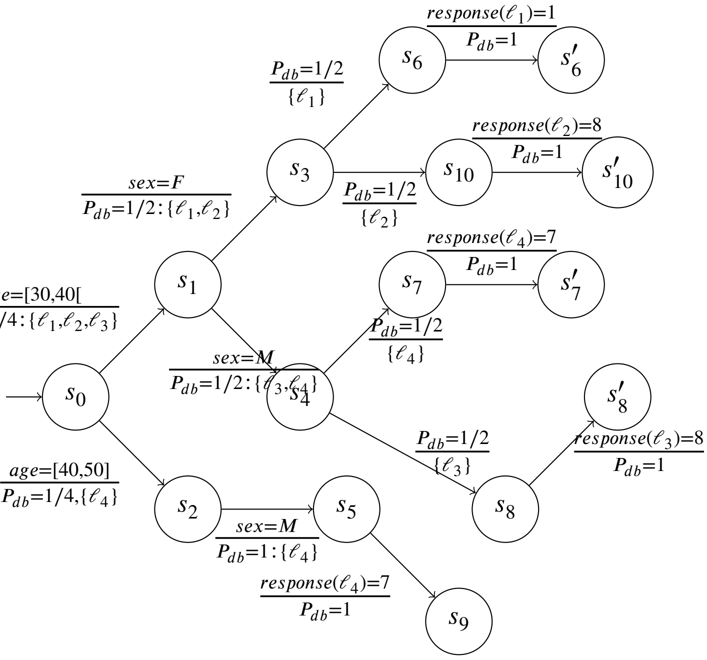


Figure 2: DLTTS-C: Basic Analyser C captures all responses

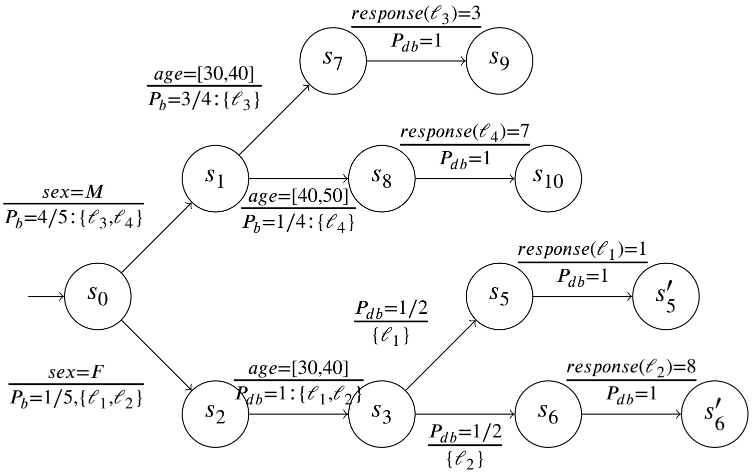


Figure 3: DLTTS-B for B's capture of responses

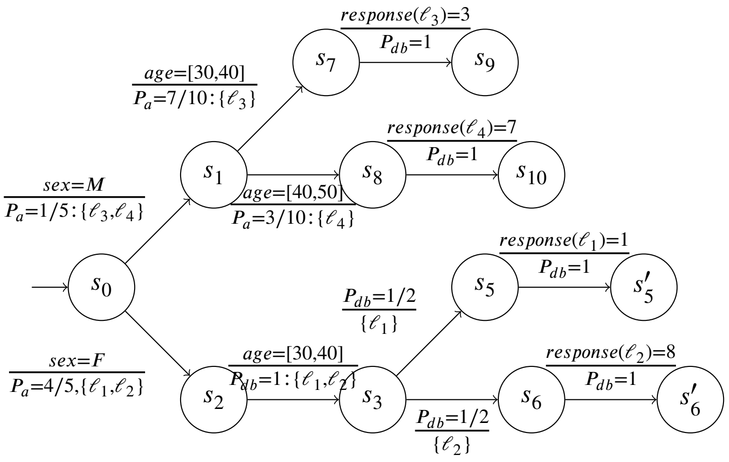


Figure 4: DLTTS-A for A's capture of responses

Attacker A: With DLTTS-A (Figure 4) Attacker A gets access to the responses of employees, preference to females of age 30 to 40. (P_a = probabilities from A's assigned objective.)

Maximal probability thresholds computed by A: Males $7/50$, Females $2/5$.
Note: The threshold $2/5$ for females is higher than the $3/16$ reported by C.

The actual probabilities for possible responses are: $Pr(response = 3|M) = 7/50$, $Pr(response = 7|M) = 3/50$, $Pr(response = 1|F) = 2/5$ and $Pr(response = 8|F) = 2/5$.

In view of our earlier remark, we may deduce from these details, that:

- B's attack succeeds at node s_7 ($response(\ell_3)$) and node s_8 ($response(\ell_4)$), on DLTTS-B.

- A's attack succeeds at node s_5 ($response(\ell_1)$) and node s_6 ($response(\ell_2)$), on DLTTS-A.

Such 'defeats' for the Basic Analyser can be avoided if the administrator of D (and the oracle mechanism $\mathcal{O}$ in the DLTTS) implement a strategy for better protecting the access to the special values in the database D ('responses', in this example). We present such a strategy in the following subsection.

8.3 A Strategy for better 'Secrecy' Protection

For any attacker N, and a DLTTS-N constructed by N modeling his/her query-runs on the given base D, if s is a node where the incoming transition has a singleton label set $\{\ell\}$, we shall denote by $Pr(s, l; N)$ the probability computed at s with $\succ$-priority transitions along the runs to s from the initial state. We shall denote by $Max_{pr}(\ell; N)$ the maximum of all these probabilities, taken over all such nodes s on DLTTS-N.

(1) Let s be a node on DLTTS-B, with a singleton $\{\ell\}$ labeling the incoming transition.
IF $Max_{pr}(\ell; C) < Pr(s, \ell; B)$,
THEN *Switch OFF* the outgoing transition *response(l)* at s.

(2) Do the same on DLTTS-A, with respect to DLTTS-C.

Details:

(i) We have: $3/16 = Max_{pr}(\ell_3; C) < Pr(s_7, \ell_3; B) = 3/5$, and $3/16 = Max_{pr}(\ell_4; C) < Pr(s_8, \ell_4; B) = 1/5$.

(i) We have: $3/16 = Max_{pr}(\ell_1; C) < Pr(s_5, \ell_1; A) = 2/5$,

and $3/16 = Max_{pr}(\ell_2; C) < Pr(s_6, \ell_2; A) = 2/5$.

It follows that the outgoing transition 'response()' can be switched OFF, if we apply the the strategy above at nodes s_7, s_8 on DLTTS-B, and at nodes s_5, s_6 on DLTTS-A. □

We are in a position now to formulate an empirical strategy for better protecting access to the 'special' values, for any database. The formulation below is for *any* general database D with a 'column of protected values' (anonymized or not) that we shall still refer to as 'responses', and *any* attacker N. It is assumed in this formulation that the administrator of the base D has made an appropriate choice for the 'Basic Analyser' C.

The Strategy:

• Let s be a node on a DLTTS-N under construction by an Attacker N for access to the responses, where the incoming transition at s on DLTTS-N has a singleton label ℓ.

• Suppose the probability $Pr(s, \ell; N)$ at s, computed along the runs to s with the supposed initial knowledge of N, and $\succ$-priority transitions all along, satisfies the condition:

$$Pr(s, \ell; N)) > Max_{pr}(\ell; C).$$

Then *switch OFF* the outgoing transition *response*(ℓ) at the node s on DLTTS-N.

9 Conclusion

As we have already mentioned earlier, our lookout for a logical formalism for privacy analysis, which could also be mingled with the notion of a 'value-wise' metric between (type compatible) atomic data, has been influenced by many works in the last couple of decades, although not with the same objective. As the developments in this work show, our *syntax*-based metric can almost directly handle data of 'mixed types', which can be numbers or literals , but can also be 'anonymized' as intervals or sets; they can also be taxonomically related to each other on a tree structure.

As has been shown in Section 7, the value-wise (partial) metric, constructed in Section 4 on type compatible sets of data, has led us to a novel and finer notion of ϵ-distinguishabilty on mechanisms answering queries. The practical application for the DLTTS vision that we have presented in Section 8 of the current paper is an addition to our earlier work [1]. Although rather simple, we believe it must be sufficiently illustrative of how the DLTTS vision can be used. It must also be noted, on the other hand, that the syntactic developments presented in Section 8 have some similarities with the semantic considerations presented in [7].

We have not considered any notions of noisy channels perturbing numerical data in the databases; but it is not difficult to extend the DLTTS setup – and the mechanism $\mathcal{M}$ answering the queries – of our work, to handle noise additions. It will then have to be assumed that the internal (saturation) procedure C, at every state in the DLTTS, incorporates the three well-known noise adding mechanisms: the Laplace, Gauss, and exponential mechanisms, with the assumption that noise additions to numerical values *is done in a bounded fashion* – as in e.g., [12] –, so as to be from a finite prescribed domain around the values. It will then have to be assumed that the tuples with noisy data are also part of the base signature $\mathcal{E}$. The notion of ϵ-local-indistinguishabilty between tuples with noisy data can also be defined in such an extended setup, under these assumptions.

As part of future work, we hope to generalize the value-wise (partial) metric constructed in Section 4 of the current paper, by assigning different 'weights' to the columns of the given base. That could be one of the techniques to 'disfavor' the columns in the database that tend to be 'noisier', or of lesser interest. That would also offer the possibility of taking into account possible dependencies between some of the columns in the database. We hope to deduce still finer notions of adjacency on databases, and of ϵ-distinguishabilty on query answering mechanisms, with such a refinement. Diverses experimentaions based on such ideas are part of the project aiming to validate the DLTTS formalism. As concerns the strategy for better secrecy protection, proposed in Section 8.3, the crucial assumption is on the appropriate choice of the (or a) 'Basic Analyser' by the system administrator; it seems a priori rather specific to the context/example considered, and notions like 'completeness' or 'soundness' of any strategy may not be very easily formalizable.

The principal goal of the DLTTS model presented in this article is to serve as an *analyser* of the knowledge that a querier (considered in this work to be an *honest-but-curious* attacker) accumulates when querying a database containing private information – usually protected by simple privacy policies represented as the values of specific tuples. We show how the DLTTS can be used as a core model by a database administrator to detect privacy breaches, and to compute the probability thresholds for these breaches; in certain cases, these thresholds can also be kept under control with simple strategies, deciding not to answer further queries in certain situations (empirically, but with certain further explanations).

References

[1] S. Anantharaman, S. Frittella, B. Nguyen. "Privacy Analysis with a Distributed Transition System and a Data-Wise Metric" In: Privacy in Statistical Databases, PARIS, France, Lecture Notes in Computer Science, Vol. PSD 2022 (LNCS 13643). Pp. 15-30, Springer, 09. 2022.

[2] G. Barthe, B. Köpf, F. Olmedo, S.Z. Béguelin. "Probabilistic relational reasoning for differential privacy". In: Proceedings of POPL, ACM (2012)

[3] G. Barthe, R. Chadha, V. Jagannath, A. Prasad Sistla, M. Viswanathan. "Deciding Differential Privacy for Programs with Finite Inputs and Outputs". In: LICS'20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020.

[4] V. Castiglioni, K. Chatzikokolakis, C. Palamidessi. "A Logical Characterization of Differential Privacy via Behavioral Metrics". In: Formal Aspects of Component Software (FACS), Pohang, South Korea. pp. 75–96, Oct. 2018.

[5] V. Castiglioni, M. Loreti, S. Tini. "The metric linear-time branching-time spectrum on nondeterministic probabilistic processes". In: Theoretical Comp. Science, Vol. 813:20–69, 2020.

[6] K. Chatzikokolakis, M. Andrés, N. Bordenabe, C. Palamidessi. "Broadening the Scope of Differential Privacy Using Metrics". In: Privacy Enhancing Technologies Symposium (PETS), Bloomington, IND (US), pp. 82–102, 2013,

[7] Y. Chen, W. W. Chu. "Database Security Protection via Cokllaborative Inference Detection". In: IEEE Transactions on Knowledge and Data Engineering, 20(8): 1013-1027 (2008).

[8] L. de Alfaro, M. Faella, M. Stoelinga. "Linear and Branching System Metrics". In: IEEE Trans. on Software Engineering, Vol. 35(2):258–273, 2009.

[9] T. Dalenius. "Findig a Needle in Haystack" (or 'Identifying Anonymous Census Records') In: J. of Official Statistics, Vol. 2 No. 3, pp. 329–336, 1986.

[10] C. Dwork. "Differential privacy". In: Proceedings of ICALP 2006. LNCS (Springer–Verlag), Vol. 4052, pp. 1–12, 2006.

[11] C. Dwork. A. Roth. “The Algorithmic Foundations of Differential Privacy”. In: Found. Trends Theor. Comput. Sci., Vol. 9:3-4, pp. 211–407, 2014.

[12] N. Holohan, S. Antonatos, S. Braghin, P. M. Aonghusa. “The Bounded Laplace Mechanism in Differential Privacy”. In: Journal of Privacy and Confidentiality (Proc. TPDP 2018), Vol. 10 (1), 2020.

[13] R. Segala. “Modeling and Verification of Randomized Distributed Real-Time Systems”. Ph.D. thesis, MIT (1995).

[14] R. Segala. “A compositional trace-based semantics for probabilistic automata”. In: Proc. CONCUR’95, 1995, pp. 234–248.

[15] R. Segala, N.A. Lynch. “Probabilistic simulations for probabilistic processes”. In: Nord. J. Comput. 2(2):250–273, 1995.

[16] Stanley L. Warner. “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias” In: Journal of the American Statistical Association Vol. 60(309), pp. 63–69, 1965.

[17] Z. Wu, M. Palmer. “Verb Semantics and Lexical selection”. In: Proc. 32nd Annual meeting of the Associations for Comp. Linguistics, pp 133-138. 1994.

10 Appendix

Taxonomies are frequent in machine learning. Data mining and clustering techniques employ reasonings based on measures of symmetry, or on metrics, depending on the objective. The Wu-Palmer symmetry measure on tree-structured taxonomies is one among those in use; it is defined as follows ([17]): Let $\mathcal{T}$ be a given taxonomy tree. For any node x on $\mathcal{T}$, define its depth c_x as the number of nodes from the root to x (both included), along the path from the root to x. For any pair x, y of nodes on $\mathcal{T}$, let c_{xy} be the depth of the common ancestor of x, y that is *farthest* from the root. The Wu-Palmer symmetry measure between the nodes x, y on $\mathcal{T}$ is then defined as $\mathrm{WP}(x, y) = \frac{2\, c_{xy}}{c_x + c_y}$. This measure, although considered satisfactory for many purposes, is known to have some disadvantages such as not being conform to semantics in several situations.

What we are interested in, for the purposes of our current paper, is a *metric* between the nodes of a taxonomy tree, which in addition will suit our semantic considerations. This is the objective of our Lemma below. (A result that seems to be unknown, to our knowledge.)

Lemma 3. *On any taxonomy tree $\mathcal{T}$, the binary function between its nodes defined by* $d_{wp}(x, y) = 1 - \frac{2\, c_{xy}}{c_x + c_y}$ (notation as above) *is a metric.*

Proof: We drop the suffix wp for this proof, and just write d. Clearly $d(x, y) = d(y, x)$; and $d(x, y) = 0$ if and only if $x = y$. We only have to prove the Triangle Inequality; i.e. show that $d(x, z) \leq d(x, y) + d(y, z)$ holds for any three nodes x, y, z on $\mathcal{T}$. A 'configuration' can be typically represented in its 'most general form' by the diagram below. The boldface characters X, Y, Z, a, h in the diagram all stand for the *number of arcs* on the corresponding paths. So that, for the depths of x, y, z, and of their farthest common ancestors on the tree, we get:

$$c_x = X + h + 1, \;\; c_y = Y + h + a + 1, \;\; c_z = Z + h + a + 1,$$
$$c_{xy} = h + 1, \;\; c_{yz} = h + a + 1, \;\; c_{xz} = h + 1$$

The '+1' in these equalities is because the X, Y, Z, a, h are the *number of arcs* on the paths, while the depths are the number of nodes. The X, Y, Z, a, h must all be integers ≥ 0. For the Triangle Inequality on the three nodes x, y, z on $\mathcal{T}$, it suffices to prove the following two relations:

$$d(x, z) \leq d(x, y) + d(y, z) \;\; \text{and} \;\; d(y, z) \leq d(y, x) + d(x, z).$$

by showing that the following two algebraic inequalities hold:

$$(1)\; 1 - \frac{2*(h+1)}{(X+Y+2*h+a+2)} + 1 - \frac{2*(h+a+1)}{(Y+Z+2*h+2*a+2)} \geq 1 - \frac{2*(h+1)}{(X+Z+2*h+a+2)}$$
$$(2)\; 1 - \frac{2*(h+1)}{(X+Y+2*h+a+2)} + 1 - \frac{2*(h+1)}{(X+Z+2*h+2*a+2)} \geq 1 - \frac{2*(h+a+1)}{(Y+Z+2*h+2*a+2)}$$

The third relation $d(x, y) \leq d(x, z) + d(z, y)$ is proved by just exchanging the roles of Y and Z in the proof of inequality (1).

Inequality (1): We eliminate the denominators (all strictly positive), and write it out as an inequality between two polynomials $eq1, eq2$ on X, Y, Z, h, a, which must be satisfied for all their non-negative integer values:

$eq1\, :\, (X + Y + 2 * h + a + 2) * (Y + Z + 2 * h + 2 * a + 2) * (X + Z + 2 * h + a + 2)$

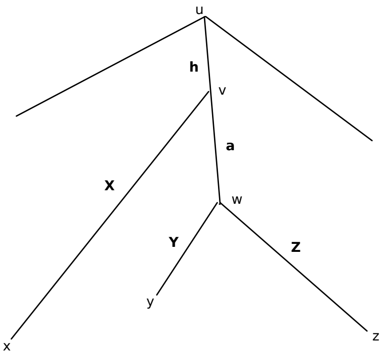
$$\begin{aligned}eq2\,:\,&(h+1)*(Y+Z+2*h+2*a+2)*(X+Z+2*h+a+2)\\&+(h+a+1)*(X+Y+2*h+a+2)*(X+Z+2*h+a+2)\\&-(h+1)*(X+Y+2*h+a+2)*(Y+Z+2*h+2*a+2)\end{aligned}$$

$eq\,:\,eq1-2*eq2.$ We need to check: $eq \geq 0$?

The equation *eq* once expanded (e.g., under *Maxima*) appears as:

$$eq\,:\,YZ^2+XZ^2+aZ^2+Y^2Z+2XYZ+4hYZ+2aYZ+4YZ+X^2Z+4hXZ+2aXZ+4XZ+a^2Z+XY^2+4hY^2+aY^2+4Y^2+X^2Y+4hXY+2aXY+4XY+8h^2Y+8ahY+16hY+a^2Y+8aY+8Y$$

The coefficients are all positive, and inequality (1) is proved.



Inequality (2): We first define the following polynomial expressions:

$$eq3\,:\,(X+Y+2*h+a+2)*(X+Z+2*h+a+2)*(Y+Z+2*h+2*a+2);$$

$$eq4\,:\,(h+1)*(Y+Z+2*h+2*a+2)*(2*X+Y+Z+4*h+2*a+4);$$

$$eq5\,:\,(h+a+1)*(X+Y+2*h+a+2)*(X+Z+2*h+a+2);$$

If we set $eqn\,:\,eq3+2*eq5-2*eq4,$ we get

$$\begin{aligned}eqn\,:\,&-2(h+1)*(Z+Y+2h+2a+2)*(Z+Y+2X+4h+2a+4)+\\&(Y+X+2h+a+2)*(Z+X+2h+a+2)(Z+Y+2h+2a+2)+\\&2(h+a+1)*(Y+X+2h+a+2)*(Z+X+2h+a+2)\end{aligned}$$

Inequality (2) is proved by showing that *eqn* remains non-negative for all non-negative values of X, Y, Z, h, a; we expand *eqn* (with *Maxima*), to get:

$$eqn\text{: } YZ^2+XZ^2+aZ^2+Y^2Z+2XYZ+4hYZ+6aYZ+4YZ+X^2Z+4hXZ+6aXZ+4XZ+8ahZ+5a^2Z+8aZ+XY^2+aY^2+X^2Y+4hXY+6aXY+4XY+8ahY+5a^2Y+8aY+4hX^2+4aX^2+4X^2+8h^2X+16ahX+16hX+8a^2X+16aX+8X+8ah^2+12a^2h+16ah+4a^3+12a^2+8a$$

The coefficients are all positive, so we are done. □