On Linear Time Decidability of Differential Privacy for Programs with Unbounded Inputs

Rohit Chadha
Email: chadhar@missouri.edu

A. Prasad Sistla
Email: sistla@uic.edu

Mahesh Viswanathan
Email: vmahesh@illinois.edu

***Abstract*—We introduce an automata model for describing interesting classes of differential privacy mechanisms/algorithms that include known mechanisms from the literature. These automata can model algorithms whose inputs can be an unbounded sequence of real-valued query answers. We consider the problem of checking whether there exists a constant d such that the algorithm described by these automata are $d\epsilon$-differentially private for all positive values of the privacy budget parameter ϵ. We show that this problem can be decided in time linear in the automaton's size by identifying a necessary and sufficient condition on the underlying graph of the automaton. This paper's results are the first decidability results known for algorithms with an unbounded number of query answers taking values from the set of reals.**

I. Introduction

Differential privacy [1], [2] is a technique developed to preserve individuals' privacy while performing statistical computations on databases containing private information. The differential privacy framework trades accuracy for privacy. In the framework, a differential privacy mechanism mediates data exchange between the database and data analyst. When the mechanism returns the answer to an analyst's query, it introduces *random* noise in the query result before forwarding it to the analyst. The mechanism is parameterized by a *privacy budget parameter* ϵ, and the noise added depends on this parameter. The privacy guarantees are also stated in terms of ϵ — a mechanism is said to be *$d\epsilon$-differentially private* if the probability of observing a given output on two adjacent databases differ only up-to a factor of $e^{d\epsilon}$, where $d > 0$ is a constant and e is the Euler's constant. Setting ϵ allows the database manager to choose the trade-off between accuracy and privacy. Intuitively, smaller values of ϵ imply improved privacy guarantees but at the cost of increased inaccuracy in the observed output.

Designing correct differential privacy mechanisms is subtle and error-prone, and even relatively minor tweaks to correct mechanisms can lead to loss of privacy as evidenced by the Sparse Vector Technique (SVT) [3], [4]. This difficulty has generated interest in formally verifying the privacy claims of differential privacy mechanisms. Verifying differential privacy is challenging for several reasons. First, the behavior of a privacy mechanism changes with ϵ as the random noise employed by the mechanism is parameterized by ϵ. The privacy guarantees are usually required to hold for all $\epsilon > 0$ to allow a manager to choose the trade-off between privacy and accuracy. Thus, the verification problem is inherently parametric. Secondly, the random noise employed by a mechanism typically samples from the continuous (or discrete) Laplace distribution. Thus, verification involves the analysis of an infinite-state stochastic model, even when inputs are constrained to come from a finite set. Finally, the mechanisms may need to process a potentially unbounded sequence of query answers, each of which may take any real value. Verification of differential privacy is known to be undecidable even when the mechanisms operate on a bounded sequence of query answers, each of which takes value from a finite domain [5].

978-1-6654-4895-6/21/$31.00

Three major directions of research seek to circumvent this challenge. The first direction aims to develop automated and semi-automated techniques to construct privacy proofs [6]–[15]. These techniques are not guaranteed to be complete and may fail to construct a proof even if the mechanism is differentially private. The second line of investigation develops automated techniques to search for privacy violations [16], [17] and searches amongst a bounded sequence of inputs. The third direction explores decision procedures for verifying differential privacy [5]. To circumvent the undecidability result, [5] considers mechanisms that sample from Laplacians only a bounded number of times and process (only) a bounded sequence of query answers, each of which is finite valued. Outputs of these mechanisms are also constrained to take values from a finite domain. The decision procedure developed in [5] converts the problem of checking differential privacy to checking the validity of first-order formulas in the theory of Reals with the exponential function. While the decidability of validity for the theory of Reals with exponential function is a longstanding open problem, formulas obtained in [5] fall into the decidable fragment identified by [18]. Unfortunately since it relies on the decision procedure for real arithmetic, the verification algorithm has very high complexity.

Contributions: In this paper, we present the first decision procedure for checking differential privacy for mechanisms that process an *unbounded* sequence of inputs, each of which may be real valued. Further, the mechanisms may also output real values in addition to values from a finite domain. In order to obtain decidability, we make two choices. First, we restrict mechanisms to those that can be modeled by a particular automata class, which we call DiP automata. Several mechanisms proposed in the literature, such as SVT and its variants [3], [4] and NumericSparse [2] can be modeled by DiP automata. Our decision procedure is sound and complete for

mechanisms modeled by such automata, and remarkably, runs in time linear in the size of the automaton. Second, we consider the following verification problem. Instead of asking whether a mechanism is $d\epsilon$ differentially private for a given constant $d > 0$ and for all $\epsilon > 0$, we ask whether there exists a constant d such that the mechanism is $d\epsilon$ differentially private for all $\epsilon > 0$. While the verification problem considered in this paper may appear to be less useful, note that a database manager can choose a lower ϵ to account for a higher d if the mechanism turns out to be differentially private. The relationship between the computation difficulty of checking $d\epsilon$-differential privacy for a given d and checking if there is some d such that a mechanism is $d\epsilon$-differentially private is unclear. For example, the decidability results in [5] do not extend to the verification problem we consider in this paper.

We briefly describe the DiP automata model introduced in this paper to model differential privacy mechanisms. A DiP automaton (DiPA) $\mathcal{A}$ takes arbitrarily long sequences of real-valued query results. Control states of $\mathcal{A}$ are classified into input and non-input states. The automaton also has a single variable x in which it can store a real value. When the automaton is in an input state, it reads an input value and generates a value, $\mathsf{insample}$, using a Laplace distribution, and compares $\mathsf{insample}$ with the stored value of x. It changes state depending on the result of comparison and outputs a value during the state transition. During the transition, it may also store the sampled value $\mathsf{insample}$ in x. When the automaton is in a non-input state, it does not read an input, but generates $\mathsf{insample}$ using constant parameters and resets x by storing $\mathsf{insample}$ in x and transitions to a new control state. The state transition's output may be either a discrete value from a finite domain or a real value. The real value could be sampled value $\mathsf{insample}$, or freshly sampled value $\mathsf{insample}'$. The mean and scaling factor of the Laplace distributions used for generating the sampled values $\mathsf{insample}$ and $\mathsf{insample}'$ are determined by the budget parameter ϵ and by constants that depend only on the state. Additionally, for input states, the input value is added to the mean.

Surprisingly, we show that the problem of checking whether a privacy mechanism, specified by a DiPA $\mathcal{A}$, is $d\epsilon$-differentially private, for some constant $d > 0$ and all $\epsilon > 0$, can be reduced to checking some syntactic graph-theoretic conditions on the finite graph "underlying"$\mathcal{A}$. These syntactic conditions are stated as the absence of certain kinds of cycles and paths (See Definition 11 on Page 10). These conditions can be checked in time linear in the graph's size by constructing the graph of strongly connected components of the "underlying"control flow graph. These conditions are independent of the scaling factors and means associated with sampling, and hence, differential privacy does not need to be re-proved if these parameters change.

Furthermore, if the privacy mechanism under consideration is differentially private, we can efficiently compute a constant d using the graph of strongly connected components, such that the mechanism is $d\epsilon$-differentially private for all values of $\epsilon > 0$. The computed d depends on the scaling parameters of states in $\mathcal{A}$ used when sampling. The computation of the constant d is once again linear, assuming constant time addition and comparison of numbers. We also observe that d computed by our algorithm for SVT and NumericSparse match those known in literature.

The proof that the given syntactic graph conditions are necessary and sufficient for differential privacy is highly non-trivial. To the best of our knowledge, these results are the first results giving efficient algorithms for checking differential privacy of interesting classes of mechanisms that process input query sequences of unbounded length, where the query values are real-valued, and the outputs may take real values.

Organization: The rest of the paper is organized as follows. Section II introduces basic notation and the setup of differential privacy. Our model of DiP automata is introduced in Section III. The main results characterizing when a DiP automata is differentially private are presented in Section IV. Because of their length, proofs of our main theorem are deferred to the Appendix. Related work is discussed in Section V. Finally we present our conclusions (Section VI). An extended abstract of this paper appeared in the 36th Annual Symposium on Logic in Computer Science (LICS 2021) [19]. This version consists of proofs omitted in [19].

II. Preliminaries

Sequences: For a set Σ, Σ^* denotes the set of all finite sequences/strings over Σ. We shall use τ to denote the empty sequence/string over Σ. For two sequences/strings $\rho, \sigma \in \Sigma^*$, we use their juxtaposition $\rho\sigma$ to indicate the sequence/string obtained by concatenating them in order. Consider $\sigma = a_0 a_1 \cdots a_{n-1} \in \Sigma^*$ (where $a_i \in \Sigma$). We use $|\sigma|$ to denote it's length n and use $\sigma[i]$ to denote its ith symbol a_i.

Sets and functions: Let $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{Q}^{\geq 0}, \mathbb{R}, \mathbb{R}^{>0}$ denote the set of natural numbers, integers, rational numbers, non-negative rationals, real numbers and positive real numbers, respectively. In addition, $\mathbb{R}_\infty$ will denote the set $\mathbb{R} \cup \{-\infty, \infty\}$, where $-\infty$ is the smallest and ∞ is the largest element in $\mathbb{R}_\infty$. For a real number $x \in \mathbb{R}$, $|x|$ denotes its absolute value, and $\mathsf{sgn}(x)$ denotes the *sign* function, i.e., $\mathsf{sgn}(x) = 0$ if $x = 0$, $\mathsf{sgn}(x) = -1$ if $x < 0$ and $\mathsf{sgn}(x) = 1$ if $x > 0$.For any partial function $f : A \hookrightarrow B$, where A, B are some sets, we let $\mathsf{dom}(f)$ be the set of $x \in A$ such that $f(x)$ is defined.

Laplace Distribution: Differential privacy mechanisms often add noise by sampling values from the *Laplace distribution*. The distribution, denoted $\mathsf{Lap}(k, \mu)$, is parameterized by two values — $k \geq 0$ which called the scaling parameter, and μ which is the mean. The probability density function of $\mathsf{Lap}(k, \mu)$, denoted $f_{k,\mu}$, is given by

$$f_{k,\mu}(x) = \frac{k}{2} e^{-k|x-\mu|}.$$

Therefore, for a random variable $X \sim \mathsf{Lap}(k, \mu)$ and $c \in \mathbb{R}$, we have

$$\mathsf{Prob}[X \leq c] = \frac{1}{2}\left[1 + \mathsf{sgn}(c-\mu)(1 - e^{-k|c-\mu|})\right].$$

Finally observe that for any $\mu_1, \mu_2 \geq 0$, $\mathsf{Lap}(k, \mu_1 + \mu_2)$ and $\mathsf{Lap}(k, \mu_1) + \mu_2$ are identically distributed.

Differential Privacy: Differential privacy [1] is a framework that enables statistical analysis of databases containing sensitive, personal information of individuals while ensuring that the privacy of individuals' information is not adversely affected by the results of the analysis. In the differential privacy framework, a randomized algorithm, M, called the *differential privacy mechanism* mediates the interaction between a (possibly dishonest) data analyst asking queries and a database D responding with answers. Queries are deterministic functions and typically include aggregate questions about the data, like the mean etc. In response to such a sequence of queries, the mechanism M will respond with a series of answers, whose value is computed using the actual answers from the database and random sampling, resulting in "noisy" answers. Thus, the differential privacy mechanism provides privacy at the cost of accuracy. Typically, the differential privacy mechanism's noisy response depends on a *privacy budget* $\epsilon > 0$.

The crucial definition of differential privacy captures the privacy guarantees of individuals in the database D. For an individual i in D, let $D \setminus \{i\}$ denote the database where i's information has been removed. A secure mechanism M ensures that for any individual i in D, and any sequence of possible outputs $\overline{o}$, the probability that M outputs $\overline{o}$ on a sequence of queries is approximately the same whether the interaction is with the database D or with $D \setminus \{i\}$. To capture this definition formally, we need to characterize the inputs on which M is required to behave similarly. Inputs to a differential privacy mechanism could be seen as answers from the database to a sequence of queries asked by the data analyst. If queries are aggregate queries, then answers to q on D and $D \setminus \{i\}$, for individual i, are likely to be away by at most 1.[1] This intuition leads to an often-used definition of *adjacency*, such as in SVT [2]–[4] and NumericSparse [2], that characterizes pairs of inputs on which the differential privacy mechanism M is expected to behave similarly.

Definition 1. Two sequences $\rho, \sigma \in \mathbb{R}^*$ are said to be *adjacent* if $|\rho| = |\sigma|$ and for each $i \leq |\rho|$, $|\rho[i] - \sigma[i]| \leq 1$.

Having defined adjacency between inputs, we are ready to formally define the notion of privacy. In response, to a sequence of inputs, a differential privacy mechanism produces a sequence of outputs from the set (say) Γ. Since a differential privacy mechanism M is a randomized algorithm, it will induce a probability distribution on Γ^*.

Definition 2 (ϵ-differential privacy)**.** A randomized algorithm M that gets as input a sequence of real numbers and produces an output in Γ^* is said to be *ϵ-differentially private* if for all measurable sets $S \subseteq \Gamma^*$ and adjacent $\rho, \sigma \in \mathbb{R}^*$ (Definition 1),

$$\mathsf{Prob}[M(\rho) \in S] \leq e^{\epsilon}\, \mathsf{Prob}[M(\sigma) \in S],$$

where e is the Euler constant.

[1]The difference in general can be a constant Δ.

Example 1. Let us look at a couple of classical differential privacy mechanisms from the literature. These will serve as running examples to motivate our definitions and highlight our results.

Input: $q[1:N]$
Output: $out[1:N]$

$\mathsf{r}_T \leftarrow \mathsf{Lap}(\frac{\epsilon}{2}, T)$
for $i \leftarrow 1$ **to** N **do**
  $\mathsf{r} \leftarrow \mathsf{Lap}(\frac{\epsilon}{4}, q[i])$
  if $\mathsf{r} \geq \mathsf{r}_T$ **then**
    $out[i] \leftarrow \top$
    exit
  else
    $out[i] \leftarrow \bot$
  end
end

Algorithm 1: SVT algorithm

Sparse Vector Technique (SVT) [3], [4] is an algorithm to answer the following question in a privacy preserving manner: Given a sequence of query answers $q[1:N]$ and threshold T, find the first index i such that $q[i] \geq T$. The algorithm is shown as Algorithm 1. It starts by sampling a value from the Laplace distribution with mean T, and stores this "noisy threshold" in the variable r_T. After that the algorithm reads query answer $q[i]$, perturbs it by sampling from the Laplace distribution with mean $q[i]$ to get r, and compares this "noisy query" r with the "noisy threshold" r_T. If $\mathsf{r} < \mathsf{r}_T$ then the algorithm outputs $\bot$ and continues by reading the next query. On the other hand, if $\mathsf{r} \geq \mathsf{r}_T$ then the algorithm outputs $\top$ and stops. This algorithm is known to be ϵ-differential private. It is worth observing that SVT is parameterized by ϵ; each value of ϵ gives us a new algorithm which is ϵ-differentially private for that particular value of ϵ.

Input: $q[1:N]$
Output: $out[1:N]$

$\mathsf{r}_T \leftarrow \mathsf{Lap}(\frac{4\epsilon}{9}, T)$
for $i \leftarrow 1$ **to** N **do**
  $\mathsf{r} \leftarrow \mathsf{Lap}(\frac{2\epsilon}{9}, q[i])$
  if $\mathsf{r} \geq \mathsf{r}_T$ **then**
    $out[i] \leftarrow \mathsf{Lap}(\frac{\epsilon}{9}, q[i])$
    exit
  else
    $out[i] \leftarrow \bot$
  end
end

Algorithm 2: Numeric Sparse algorithm

Consider Algorithm 2 which shows a differential privacy mechanism called Numeric Sparse [2]. The problem solved by this algorithm is very similar to the one solved by SVT (Algorithm 1) — given a sequence of query answers $q[1:N]$

and threshold T, find the first index i such that $q[i] \geq T$ *and output* $q[i]$. Algorithm 2 is similar to Algorithm 1. The only difference is that instead of outputting $\top$ when $\mathsf{r} \geq \mathsf{r}_T$, it outputs a perturbed value of $q[i]$. This algorithm is also known to be ϵ-differentially private for each possible assignment of value to ϵ.

III. DiP automata

DiP (**Di**fferentially **P**rivate) automata (DiPA for short) are a simple model to describe some differential privacy mechanisms known in the literature. Some of the features we hope to capture are those highlighted by Algorithms 1 and 2. Recall that the input to a differential privacy mechanism is a sequence of real numbers that correspond to answers to queries. The differential privacy mechanism is a randomized algorithm that processes this input, samples values from distributions like Laplace, and produces a sequence of values as output. These outputs could include real numbers (Algorithm 2). Further, as observed in Example 1, the behavior of the mechanism depends on the privacy budget ϵ. DiP automata are a formal model that have these features.

A. Syntax

A DiPA is a *parametric* automaton with finitely many control states and three real-valued variables $\mathsf{insample}, \mathsf{insample}'$ and x. While the variables $\mathsf{insample}$ and $\mathsf{insample}'$ are freshly sampled in each step, the variable x can store real values to be used in later steps. The value of the parameter ϵ (the privacy budget) influences the distribution from which the real values are sampled during an execution. The input to such an automaton is a finite sequence of real numbers. In each step the automaton does the following.

1) It samples two values, called $\mathsf{insample}$ and $\mathsf{insample}'$, drawn from the distributions $\mathsf{Lap}(d\epsilon, \mu)$ and $\mathsf{Lap}(d'\epsilon, \mu')$, respectively. The scaling factors d, d' and means μ, μ' of these distributions depend on the current state.
2) Depending on the current state, the automaton will either read a real number from the input, or not read anything from the input. If an input value a is read, then $\mathsf{insample}$ and $\mathsf{insample}'$ are updated by adding a to them.
3) The transition results in changing the control state and outputting a value. The value output could either be a symbol from a finite set (like $\perp/\top$ in Algorithm 1) or one of the two real numbers $\mathsf{insample}$ and $\mathsf{insample}'$ that are sampled in this step (like in Algorithm 2). If an input value is read then the transition could be guarded by the result of comparing the sampled value $\mathsf{insample}$ and the stored value x. It is possible that for certain values of x and $\mathsf{insample}$, no transition is enabled from the current state. In such a case, the computation ends.
4) Finally, the automaton may choose to store the sampled value $\mathsf{insample}$ in x.

The above intuition is captured by the formal definition of DiPA below and its semantics described later in this section.

Definition 3 (DiPA)**.** Let C be the set of *guard conditions* $\{\mathsf{true}, \mathsf{insample} \geq \mathsf{x}, \mathsf{insample} < \mathsf{x}\}$. A *DiP automaton* $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$ where

- Q is a finite set of states partitioned into two sets: the set of input states Q_{in} and the set of non-input states Q_{non},
- $\Sigma = \mathbb{R}$ is the input alphabet,
- Γ is a finite output alphabet,
- $q_{\mathsf{init}} \in Q$ is the initial state,
- $X = \{\mathsf{x}, \mathsf{insample}, \mathsf{insample}'\}$ is the set of variables,
- $P : Q \to \mathbb{Q}^{\geq 0} \times \mathbb{Q} \times \mathbb{Q}^{\geq 0} \times \mathbb{Q}$ is the parameter function that assigns to each state a 4-tuple (d, μ, d', μ'), where $\mathsf{insample}$ is sampled from $\mathsf{Lap}(d\epsilon, \mu)$ and $\mathsf{insample}'$ is sampled from $\mathsf{Lap}(d'\epsilon, \mu')$,
- and $\delta : (Q \times C) \hookrightarrow (Q \times (\Gamma \cup \{\mathsf{insample}, \mathsf{insample}'\}) \times \{\mathsf{true}, \mathsf{false}\})$ is the transition (partial) function that given a current state and result of comparing x with $\mathsf{insample}$, determines the next state, the output, and whether x should be updated to store $\mathsf{insample}$. The output could either be a symbol from Γ or the values $\mathsf{insample}$ and $\mathsf{insample}'$ that were sampled.

The transition function δ of a DiPA will satisfy the following four conditions.

Determinism: For any state $q \in Q$, if $\delta(q, \mathsf{true})$ is defined then $\delta(q, \mathsf{insample} \geq \mathsf{x})$ and $\delta(q, \mathsf{insample} < \mathsf{x})$ are undefined.

Output Distinction: For any state $q \in Q$, if $\delta(q, \mathsf{insample} \geq \mathsf{x})$ is defined to be (q_1, o_1, b_1) and $\delta(q, \mathsf{insample} < \mathsf{x})$ is defined to be (q_2, o_2, b_2) then $o_1 \neq o_2$, i.e., distinct transitions from a state have different outputs. Further at least one out of o_1 and o_2 belongs to Γ, i.e., both transitions cannot output real values.

Initialization: The initial state q_{init} has only one outgoing transition of the form $\delta(q_{\mathsf{init}}, \mathsf{true}) = (q, o, \mathsf{true})$ where q is a state and o is an output symbol. In other words, the guard of the first transition is always true and the first value sampled is stored in x.

Non-input transition: From any $q \in Q_{\mathsf{non}}$, if $\delta(q, c)$ is defined, then $c = \mathsf{true}$; that is, there is at most one transition from a non-input state which is always enabled.

It is useful to classify transitions of a DiPA into different types. Consider a transition $\delta(q, c) = (q', o, b)$. If $q \in Q_{\mathsf{in}}$ then it is an *input transition* and if $q \in Q_{\mathsf{non}}$ then it is a *non-input transition*. If $b = \mathsf{true}$ then the transition will set $\mathsf{x} = \mathsf{insample}$, and hence it is called an *assignment transition*. On the other hand, if $b = \mathsf{false}$, the transition will be said to be a *non-assignment* transition. A *pure assignment* transition is an assignment transition with $c = \mathsf{true}$. The initialization condition says that the (only) transition out of the initial state of a DiPA is a pure assignment transition.

Example 2. The differential privacy mechanisms in Example 1 can be modeled as DiP automata. These are shown in Fig. 1 and 2. When drawing DiPAs in this paper, we will follow these conventions. Input states will be represented as circles, while non-input states with be shown as rectangles.

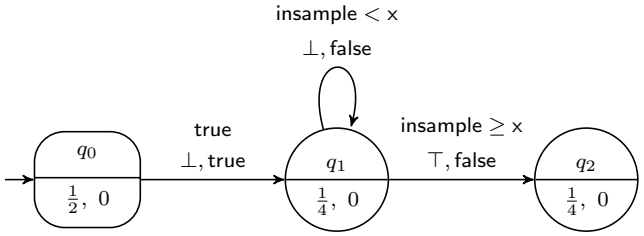

Fig. 1. DiPA $\mathcal{A}_{\mathsf{SVT}}$ modeling Algorithm 1. Threshold for the algorithm is 0 (mean for sampling insample in state q_0).

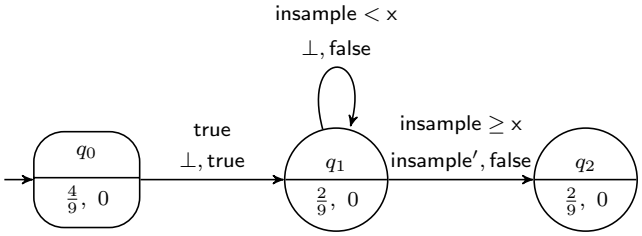

Fig. 2. DiPA $\mathcal{A}_{\mathsf{NumSp}}$ modeling Algorithm 2. The threshold is taken to be 0. Label of each state below the line shows the parameters for sampling insample. Parameters for sampling insample′ are not shown in the figure; they are $\frac{1}{9}$ (scaling factor) and 0 (mean) in every state.

The name of each state is written above the line, while the scaling factor d and mean μ of the distribution used to sample insample is written below the line. The parameters d' and μ' for sampling insample′ are not shown in the figures, but are mentioned in the caption and text when they are important; they are relevant only when insample′ is output on a transition. Edges will be labeled with the guard of the transition, followed by the output, and a Boolean to indicate whether the transition is an assignment transition.

The SVT algorithm (Algorithm 1) can be modeled as a DiPA $\mathcal{A}_{\mathsf{SVT}}$ shown in Fig. 1. Since $\mathcal{A}_{\mathsf{SVT}}$ does not output insample′ in any transition, the parameters used for sampling insample′ are not relevant. In this representation of SVT, the threshold used for comparison in the algorithm is hard-coded in the automaton as the mean parameter of the initial state q_0. In fact, without loss of generality we can take this to be 0 as shown in Fig. 1. The initial state q_0 of the automaton is a non-input state with $d = \frac{1}{2}$ and $\mu = 0$ (the threshold for the algorithm). From q_0, the algorithm samples a value that corresponds to the perturbed threshold and stores this in variable x. In state q_1, in each step it reads a query value (input), perturbs it by sampling, and compares this with the perturbed threshold stored in variable x. If the sampled value is less that x it stays in q_1, outputs $\perp$ and leaves x unchanged. On the other hand, if insample $\geq$ x then it outputs $\top$, and transitions to a terminal state q_2.

$\mathcal{A}_{\mathsf{SVT}}$ can be used to illustrate our classification of transitions. The transition from q_0 to q_1 is the only non-input transition and the only assignment transition in the automaton; all other transitions are non-assignment, input transitions. In addition, the transition from q_0 to q_1 is also a pure assignment transition, since the guard is true.

Automaton $\mathcal{A}_{\mathsf{NumSp}}$ modeling Numeric Sparse (Algorithm 2) is shown in Fig. 2. As in the case of $\mathcal{A}_{\mathsf{SVT}}$ (Fig. 1), the threshold is hard-coded in the automaton and is taken to be 0 (without loss of generality). Parameters used to sample insample′ are not shown in diagram depicting $\mathcal{A}_{\mathsf{NumSp}}$. We take those to be just be $\frac{1}{9}$ (scaling factor) and 0 (mean) in every state; in fact, these parameters for insample′ are only important for state q_1. The automaton is very similar to $\mathcal{A}_{\mathsf{SVT}}$ (Fig. 1) with the only differences being the parameters used when sampling in each state, and the fact that insample′ is output on the transition from q_1 to q_2 instead of $\top$.

B. Paths and executions

A DiPA $\mathcal{A}$ defines a probability measure on the *executions* or *paths* of $\mathcal{A}$ (henceforth just called a path). Informally, a path is just a sequence of transitions taken by the automaton. Observe that the condition of output distinction ensures that knowing the current state and output, determines which transition is taken. The input read determines the value of insample and insample′, and therefore, to define the probability of a path, we need to know the inputs read as well. Finally, on transitions where either insample or insample′ are output, to define a meaningful measure space, we need to associate an interval (v, w) in which the output value lies. Because of these reasons, we define a path to be one that describes the sequence of (control) states the automaton goes through and the sequence of inputs read and outputs produced.

Before defining a path formally, it is useful to introduce the following notation. For a pair of states $p, q \in Q$, $a \in \Sigma \cup \{\tau\}$ and $o \in \Gamma \cup (\{\mathsf{insample}, \mathsf{insample}'\} \times \mathbb{R}_\infty \times \mathbb{R}_\infty)$, we say $p \xrightarrow{a,o} q$ if $a = \tau$ whenever $p \in Q_{\mathsf{non}}$ and $a \in \Sigma$ whenever $p \in Q_{\mathsf{in}}$, and one of the following two conditions holds.

- If $o \in \Gamma$ then there is a guard $c \in C$ and Boolean $b \in \{\mathsf{true}, \mathsf{false}\}$ such that $\delta(p, c) = (q, o, b)$.
- If o is of the form (y, v, w) where $y \in \{\mathsf{insample}, \mathsf{insample}'\}$ and $v, w \in \mathbb{R}_\infty$ then there is a guard $c \in C$ and Boolean $b \in \{\mathsf{true}, \mathsf{false}\}$ such that $\delta(p, c) = (q, y, b)$. Intuitively, an "output" of the form $(\mathsf{insample}, v, w)$ (or $(\mathsf{insample}', v, w)$) indicates that the value of insample (insample′) was output in the transition and the result was a number in the interval (v, w).

The *unique* transition, or rather the quintuple (p, c, q, o', b), that witnesses $p \xrightarrow{a,o} q$ will be denoted by $\mathsf{trans}(p \xrightarrow{a,o} q)$.

Definition 4 (Path)**.** Let $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$ be a DiPA. An *execution* or *path* ρ of $\mathcal{A}$ is a sequence of the form

$$\rho = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$$

where $q_i \in Q$ for $0 \leq i \leq n$, $a_j \in \Sigma \cup \{\tau\}$ and $o_j \in \Gamma \cup (\{\mathsf{insample}, \mathsf{insample}'\} \times \mathbb{R}_\infty \times \mathbb{R}_\infty)$ for $0 \leq j < n$. In addition, we require that $q_j \xrightarrow{a_j,o_j} q_{j+1}$ for all $0 \leq j < n$.

Such a path ρ is said to be from state q_0 ($\mathsf{first}(\rho)$) to state q_n ($\mathsf{last}(\rho)$). Its *length* (denoted $|\rho|$) is the number of transitions,

namely, n. If the starting state and ending state of a path are the same (i.e., $q_0 = q_n$) and $|\rho| > 0$ then ρ is said to be a *cycle*.

It will be convenient to introduce some notation associated with paths.

Notation. Let us consider a path

$$\rho = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$$

of length n. If $|\rho| > 0$, then the *tail* of ρ, denoted $\mathsf{tail}(\rho)$, is the path of length $n-1$ given by

$$\mathsf{tail}(\rho) = q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n.$$

The ith state of the path is $\mathsf{state}(\rho[i]) = q_i$ and the ith transition is $\mathsf{trans}(\rho[i]) = \mathsf{trans}(q_i \xrightarrow{a_i,o_i} q_{i+1})$. The guard of the ith transition is $\mathsf{guard}(\rho[i]) = c$, where $\mathsf{trans}(\rho[i]) = (q_i, c, q_{i+1}, o', b)$.

Finally, it will be useful to introduce notation for the sequence of inputs read and outputs produced in a path. The output produced will be an element of $(\Gamma \cup (\mathbb{R}_\infty \times \mathbb{R}_\infty))^*$ that ignores the variable name that was output when a real value is output. For $o \in \Gamma$, define $\langle o \rangle = o$, and for o of the form (y, v, w) where $y \in \{\mathsf{insample}, \mathsf{insample}'\}$ and $v, w \in \mathbb{R}_\infty$, define $\langle o \rangle = (v, w)$.

$$\mathsf{inseq}(\rho) = a_0 a_1 \cdots a_{n-1}$$
$$\mathsf{outseq}(\rho) = \langle o_0 \rangle \langle o_1 \rangle \cdots \langle o_{n-1} \rangle$$

Two paths ρ_1 and ρ_2 will be said to be *equivalent* if they only differ in the sequence of inputs read. In other words, equivalent paths are of the same length, go through the same states, and produce the same outputs (and hence take the same transitions).

Thanks to output distinction, two paths are equivalent if and only if they start from the same state and have the same output sequences. Thus, paths are uniquely determined by starting state, input and output sequences. Finally, modifying the values input in a path yields an equivalent path.

Proposition 1. *Let ρ_1 and ρ_2 be paths of a DiPA $\mathcal{A}$ starting from the same state.*

- *ρ_1 and ρ_2 are equivalent if and only if* $\mathsf{outseq}(\rho_1) = \mathsf{outseq}(\rho_2)$.
- *If* $\mathsf{inseq}(\rho_1) = \mathsf{inseq}(\rho_2)$ *and* $\mathsf{outseq}(\rho_1) = \mathsf{outseq}(\rho_2)$ *then* $\rho_1 = \rho_2$.
- *For any sequence of reals $\overline{a} \in \Sigma^*$ such that $|\overline{a}| = |\mathsf{inseq}(\rho_1)|$, there is a unique path ρ_3 equivalent to ρ_1 such that* $\mathsf{inseq}(\rho_3) = \overline{a}$.

C. Path probabilities

We will now formally define what the probability of each path is. Recall that in each step, the automaton samples two values from Laplace distributions, and if the transition is from an input state, it adds the read input value to the sampled values and compares the result with the value stored in x. The step also outputs a value, and if the value output is one of the two sampled values, the path requires it to belong to the interval that labels the transition. The probability of such a transition thus is the probability of drawing a sample that satisfies the guard of the transition and (if the output is a real value) producing a number that lies in the interval in the output label. This intuition is formalized in a precise definition.

Let us fix a path

$$\rho = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$$

of DiPA $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$. Recall that the parameters to the Laplace distribution in each step depend on the privacy budget ϵ. In addition, the value stored in the variable x at the start of ρ influences the behavior of $\mathcal{A}$. Thus, the probability of path ρ depends on both the value for ϵ and the value of x at the start of ρ; we will denote this probability as $\mathsf{Pr}[\epsilon, x, \rho]$, where x is the initial value of x. We define this inductively on $|\rho|$. For any ϵ and any path ρ with $|\rho| = 0$, $\mathsf{Pr}[\epsilon, x, \rho] = 1$.

For a path ρ of non-zero length, let $(q_0, c, q_1, o_0, b) = \mathsf{trans}(q_0 \xrightarrow{a_0,o_0} q_1)$ be the 0th transition of ρ. Let $P(q_0) = (d, \mu, d', \mu')$ and let $\langle a_0 \rangle = a_0$ if $a_0 \in \mathbb{R}$ and $\langle a_0 \rangle = 0$ if $a_0 = \tau$. We will define constants ℓ and u as follows. If $o_0 \in \Gamma$ then $\ell = -\infty$ and $u = \infty$. Otherwise, o_0 is of the form (y, v, w) where $y \in \{\mathsf{insample}, \mathsf{insample}'\}$, and then we take $\ell = v$ and $u = w$. We assume that any integral of the form $\int_e^f g(y)dy = 0$ when $e > f$. Finally, when o_0 is of the form (y, v, w) where $y \in \{\mathsf{insample}, \mathsf{insample}'\}$ (i.e., $o_0 \notin \Gamma$), define

$$k = \int_v^w \frac{d\epsilon}{2} e^{-d\epsilon|z-\mu-\langle a_0 \rangle|} dz$$
$$k' = \int_v^w \frac{d'\epsilon}{2} e^{-d'\epsilon|z-\mu'-\langle a_0 \rangle|} dz$$

The function $\mathsf{Pr}[\cdot]$ is defined based on what c and b are. Let us fix $\nu = \mu + \langle a_0 \rangle$. We begin by considering the case when the 0th transition of ρ is a non-assignment transition, i.e., when $b = \mathsf{false}$.

- **Case** $c = \mathsf{true}$**:** If $o_0 \in \Gamma$ then $\mathsf{Pr}[\epsilon, x, \rho] = \mathsf{Pr}[\epsilon, x, \mathsf{tail}(\rho)]$. If $o_0 = (\mathsf{insample}, v, w)$ then $\mathsf{Pr}[\epsilon, x, \rho] = k\mathsf{Pr}[\epsilon, x, \mathsf{tail}(\rho)]$ and if $o_0 = (\mathsf{insample}', v, w)$ then $\mathsf{Pr}[\epsilon, x, \rho] = k'\mathsf{Pr}[\epsilon, x, \mathsf{tail}(\rho)]$
- **Case** $c = \mathsf{insample} \geq \mathsf{x}$**:** If o_0 is of the form $(\mathsf{insample}', v, w)$ (i.e., $\mathsf{insample}'$ is output) then

  $$\mathsf{Pr}[\epsilon, x, \rho] = k' \left(\int_x^\infty \frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} dz \right) \mathsf{Pr}[\epsilon, x, \mathsf{tail}(\rho)].$$

  Otherwise, taking $\ell' = \max(x, \ell)$,

  $$\mathsf{Pr}[\epsilon, x, \rho] = \left(\int_{\ell'}^u \frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} dz \right) \mathsf{Pr}[\epsilon, x, \mathsf{tail}(\rho)].$$

- **Case** $c = \mathsf{insample} < \mathsf{x}$**:** If o_0 is of the form $(\mathsf{insample}', v, w)$ (i.e., $\mathsf{insample}'$ is output) then

  $$\mathsf{Pr}[\epsilon, x, \rho] = k' \left(\int_{-\infty}^x \frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} dz \right) \mathsf{Pr}[\epsilon, x, \mathsf{tail}(\rho)].$$

  Otherwise, taking $u' = \min(x, u)$,

  $$\mathsf{Pr}[\epsilon, x, \rho] = \left(\int_\ell^{u'} \frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} dz \right) \mathsf{Pr}[\epsilon, x, \mathsf{tail}(\rho)].$$

Next, when the 0th transition of ρ is an assignment transition, i.e., $b = \mathsf{true}$, $\mathsf{Pr}[\cdot]$ is defined as follows.

- **Case** $c = \mathsf{true}$**:** If o_0 is of the form $(\mathsf{insample}', v, w)$ (i.e., $\mathsf{insample}'$ is output) then

$$\mathsf{Pr}[\epsilon, x, \rho] = k' \int_{-\infty}^{\infty} \left(\frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} \right) \mathsf{Pr}[\epsilon, z, \mathsf{tail}(\rho)] dz.$$

  Otherwise,

$$\mathsf{Pr}[\epsilon, x, \rho] = \int_{\ell}^{u} \left(\frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} \right) \mathsf{Pr}[\epsilon, z, \mathsf{tail}(\rho)] dz.$$

- **Case** $c = \mathsf{insample} \geq \mathsf{x}$**:** If o_0 is of the form $(\mathsf{insample}', v, w)$ (i.e., $\mathsf{insample}'$ is output) then

$$\mathsf{Pr}[\epsilon, x, \rho] = k' \int_{x}^{\infty} \left(\frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} \right) \mathsf{Pr}[\epsilon, z, \mathsf{tail}(\rho)] dz.$$

  Otherwise, taking $\ell' = \max(x, \ell)$,

$$\mathsf{Pr}[\epsilon, x, \rho] = \int_{\ell'}^{u} \left(\frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} \right) \mathsf{Pr}[\epsilon, z, \mathsf{tail}(\rho)] dz.$$

- **Case** $c = \mathsf{insample} < \mathsf{x}$**:** If o_0 is of the form $(\mathsf{insample}', v, w)$ (i.e., $\mathsf{insample}'$ is output) then

$$\mathsf{Pr}[\epsilon, x, \rho] = k' \int_{-\infty}^{x} \left(\frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} \right) \mathsf{Pr}[\epsilon, z, \mathsf{tail}(\rho)] dz.$$

  Otherwise, taking $u' = \min(u, x)$,

$$\mathsf{Pr}[\epsilon, x, \rho] = \int_{\ell}^{u'} \left(\frac{d\epsilon}{2} e^{-d\epsilon|z-\nu|} \right) \mathsf{Pr}[\epsilon, z, \mathsf{tail}(\rho)] dz.$$

We will abuse notation and use $\mathsf{Pr}[\cdot]$ to also refer to $\mathsf{Pr}[x, \rho] = \lambda\epsilon.\ \mathsf{Pr}[\epsilon, x, \rho]$. Notice that when ρ starts from q_{init}, because of the initialization condition of DiPA, the value of $\mathsf{Pr}[\cdot]$ does not depend on the initial value of x. For such paths, we may drop the initial value of x from the argument list of $\mathsf{Pr}[\cdot]$ to reduce notational overhead. Even though we plan to use the same function name, the number of arguments to $\mathsf{Pr}[\cdot]$ will disambiguate what we mean.

Example 3. Let use consider the DiPA $\mathcal{A}_{\mathsf{SVT}}$ shown in Fig. 1. A couple of example paths of the automaton are the following.

$$\rho_1 = q_0 \xrightarrow{\tau,\perp} q_1 \xrightarrow{0,\perp} q_1 \xrightarrow{1,\top} q_2$$
$$\rho_2 = q_0 \xrightarrow{\tau,\perp} q_1 \xrightarrow{1,\perp} q_1 \xrightarrow{1,\top} q_2$$

Paths ρ_1 and ρ_2 only differ in the inputs they read: $\mathsf{inseq}(\rho_1) = \tau \cdot 0 \cdot 1 = 01$, while $\mathsf{inseq}(\rho_2) = 11$. Thus, ρ_1 and ρ_2 are equivalent paths. Notice that ρ_1 and ρ_2 are adjacent (Definition 1). The outputs produced in these executions is given by $\mathsf{outseq}(\rho_1) = \mathsf{outseq}(\rho_2) = \perp\perp\top$.

Let us now consider $\mathsf{Pr}[\epsilon, 0, \rho_1]$. Since the transition out of q_0 is a pure assignment transition, the initial value of x (namely 0 in this example) does not influence the value of $\mathsf{Pr}[\epsilon, 0, \rho_1]$. Let X_T, X_1, X_2 be random variables where $X_T \sim \mathsf{Lap}(\frac{\epsilon}{2}, 0)$, $X_1 \sim \mathsf{Lap}(\frac{\epsilon}{4}, 0) + 0$, and $X_2 \sim \mathsf{Lap}(\frac{\epsilon}{4}, 0) + 1$. We can see that

$$\mathsf{Pr}[\epsilon, 0, \rho_1] = \mathsf{Prob}[X_1 < X_T \ \wedge \ X_2 \geq X_T].$$

Based on how the random variables are distributed, this can be calculated to be

$$\mathsf{Prob}[X_1 < X_T \ \wedge \ X_2 \geq X_T] = \frac{24e^{\frac{3\epsilon}{4}} - 1 + 8e^{\frac{\epsilon}{4}} - 21e^{\frac{\epsilon}{2}}}{48e^{\frac{3\epsilon}{4}}}.$$

The calculation of $\mathsf{Pr}[\epsilon, 0, \rho_2]$ is similar. Let X_1' be the random variable with $X_1' \sim \mathsf{Lap}(\frac{\epsilon}{4}, 0) + 1$. Then the desired probability is same as $\mathsf{Prob}[X_1' < X_T \ \wedge X_2 \geq X_T]$. This can be calculated to be

$$\begin{aligned} \mathsf{Pr}[\epsilon, 0, \rho_2] &= \mathsf{Prob}[X_1' < X_T \ \wedge \ X_2 \geq X_T] \\ &= \tfrac{-22 + 32e^{\frac{\epsilon}{4}} - 3\epsilon}{48e^{\frac{\epsilon}{2}}}. \end{aligned}$$

The focus of this paper is to study the computational problem of checking differential privacy for DiP automata. We conclude this section with a precise definition of this problem. In order to do that we first specialize the definition of differential privacy to the setting of DiPA.

Thanks to Proposition 1, the definition of ϵ-differential privacy [2] specializes to the following in the case of DiPA.

Definition 5. A DiPA $\mathcal{A}$ is $d\epsilon$-differentially private (for $d > 0$, $\epsilon > 0$) iff for every pair of *equivalent* paths ρ_1, ρ_2 starting from the initial state such that $\mathsf{inseq}(\rho_1)$ and $\mathsf{inseq}(\rho_2)$ are *adjacent* [2],

$$\mathsf{Pr}[\epsilon, \rho_1] \leq e^{d\epsilon}\ \mathsf{Pr}[\epsilon, \rho_2].$$

Differential Privacy Problem: Given a DiPA $\mathcal{A}$ (with privacy parameter ϵ), determine if there is a $d > 0$ such that for every $\epsilon > 0$, $\mathcal{A}$ is $d\epsilon$-differentially private.

IV. Deciding Differential Privacy

The central computational problem that this paper studies is the following: Given a DiPA $\mathcal{A}$ determine if there is a $d > 0$ such that for all $\epsilon > 0$, $\mathcal{A}$ is $d\epsilon$-differentially private. In this section we present the main result of this paper, namely, that this problem is efficiently decidable in linear time. We also show that we can compute an upper bound on d in linear time if $\mathcal{A}$ is differentially private. The crux of the proof is the identification of simple graph-theoretic conditions that are both *necessary and sufficient* to ensure a DiPA is $d\epsilon$-differentially private for all ϵ and some d.

Before presenting the properties that are needed to guarantee differential privacy, we first define the notion of reachability. Let us fix a DiPA $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$. A state q is said to be *reachable* if there is a path ρ starting from state q_{init} and ending in q. In addition, we say that a path (cycle) ρ is reachable if there is a path ρ' from q_{init} to $\mathsf{first}(\rho)$. We now start by identifying the first interesting property.

Definition 6. A path ρ in a DiPA $\mathcal{A}$ is said to be a *leaking path* if there exist indices i, j with $0 \leq i < j < |\rho|$ such that the ith transition $\mathsf{trans}(\rho[i])$ is an assignment transition and the guard of the jth transition $\mathsf{guard}(\rho[j]) \neq \mathsf{true}$. A leaking path ρ is said to be a *leaking cycle* if it is also a cycle.

[2] See Definition 1 on Page 3.

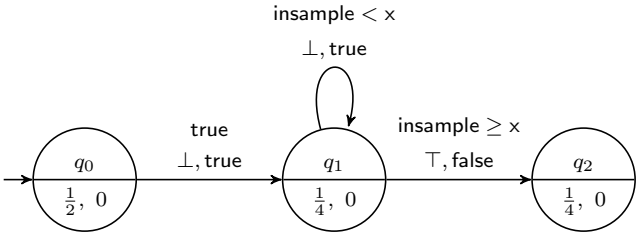

Fig. 3. DiPA $\mathcal{A}_{\mathsf{sort}}$ modeling an algorithm that checks whether the sequence of real numbers given as input are sorted in descending order. Since $\mathsf{insample}'$ is not output in any state, the parameters used in sampling $\mathsf{insample}'$ are not important.

Intuitively, in a leaking path, the variable x is assigned a value in some transition which is used in the guard of a later transition. Observe that if a path is leaking, then all paths equivalent to it are also leaking. The presence of a reachable leaking cycle is a witness that the DiPA is not differentially private. The intuition behind this is as follows. One can show that there are a pair of adjacent inputs such that traversing the leaking cycle C once on these inputs results in two paths, the ratio of whose probabilities is at least $e^{k\epsilon}$ for some number k for sufficiently large ϵ. Thus, given d, we can find an ℓ and an ϵ such that traversing the cycle ℓ times "exhausts the privacy budget", i.e., the adjacent inputs corresponding to these ℓ repetitions have probabilities whose ratio is at least $e^{d\epsilon}$. We illustrate this through our next example.

Example 4. Consider an algorithm that checks whether the input sequence of real numbers is sorted in descending order. The goal of the algorithm is to read a sequence of numbers, output $\perp$ as long as it is sorted, and output $\top$ the first time it encounters two numbers in the wrong order and stop. A "differentially private" version of this algorithm is modeled by DiPA $\mathcal{A}_{\mathsf{sort}}$ shown in Fig. 3. It works as follows. It starts by reading an input in state q_0, perturbing it by sampling from the Laplace distribution, outputting $\perp$, and storing the perturbed input in x. In state q_1, $\mathcal{A}_{\mathsf{sort}}$ repeatedly reads an input, perturbs it, and checks if it is less than the previous perturbed value read by the automaton, which is stored in x. If it is, the automaton outputs $\perp$, saves the new perturbed value, and stays in q_1 to read the next input symbol. On the other hand, if the new value is greater, then it outputs $\top$ and moves to a terminal state. $\mathcal{A}_{\mathsf{sort}}$ is almost identical to the automaton $\mathcal{A}_{\mathsf{SVT}}$ (Fig. 1) — the only difference is that initial state of $\mathcal{A}_{\mathsf{sort}}$ is an input state as opposed to a non-input state, and the self loop on state q_1 is an assignment transition.

This difference (that the self loop on q_1 is an assignment transition) turns out to be critical; $\mathcal{A}_{\mathsf{sort}}$ is not differentially private even though $\mathcal{A}_{\mathsf{SVT}}$ is. Observe that the cycle $q_1 \xrightarrow{a_0,\perp} q_1 \xrightarrow{a_1,\perp} q_1$ is a leaking cycle as the 0th transition is an assignment transition and the 1st transition's guard is $\mathsf{insample} < \mathsf{x}$. We can exploit this cycle to demonstrate why $\mathcal{A}_{\mathsf{sort}}$ is not differentially private. Consider the paths of length n given as

$$\rho_1^n = q_0 \xrightarrow{0,\perp} q_1 \xrightarrow{-1,\perp} q_1 \xrightarrow{-2,\perp} q_1 \xrightarrow{-3,\perp} q_1 \xrightarrow{-4,\perp} q_1 \cdots$$
$$\rho_2^n = q_0 \xrightarrow{0,\perp} q_1 \xrightarrow{-2,\perp} q_1 \xrightarrow{-1,\perp} q_1 \xrightarrow{-4,\perp} q_1 \xrightarrow{-3,\perp} q_1 \cdots$$

Observe that for all n, $\mathsf{inseq}(\rho_1^n)$ and $\mathsf{inseq}(\rho_2^n)$ are adjacent (Definition 1). Moreover, for any $d > 0$, there is an n and ϵ, such that the ratio of $\Pr[\epsilon, \rho_1^n]$ and $\Pr[\epsilon, \rho_2^n]$ is $> e^{d\epsilon}$. Thus, $\mathcal{A}$ is not $d\epsilon$-differentially private for any d.

Absence of a leaking cycle does not guarantee differential privacy. Privacy leaks can occur with other types of paths and cycles. We define one such path next.

Definition 7. A cycle ρ of a DiPA $\mathcal{A}$ is called an L-cycle (respectively, G-cycle) if there is an $i < |\rho|$ such that $\mathsf{guard}(\rho[i]) = \mathsf{insample} < \mathsf{x}$ (respectively, $\mathsf{guard}(\rho[i]) = \mathsf{insample} \geq \mathsf{x}$).

We say that a path ρ of a DiPA $\mathcal{A}$ is an AL-path (respectively, AG-path) if all assignment transitions on ρ have guard $\mathsf{insample} < \mathsf{x}$ (respectively, $\mathsf{insample} \geq \mathsf{x}$).

Observe that a cycle can be both an L-cycle and a G-cycle. Further, a path with no assignment transitions (including the empty path) is simultaneously both an AL-path and an AG-path.

Definition 8. A pair of cycles (C, C') in a DiPA $\mathcal{A}$ is called a *leaking pair* if one of the following two conditions is satisfied.

1) C is an L-cycle, C' is a G-cycle and there is an AG-path from a state in C to a state in C'.
2) C is a G-cycle, C' is an L-cycle and there is an AL-path from a state in C to a state in C'.

Observe that if C is an L-cycle as well as a G-cycle, then the pair (C, C) is a leaking pair with the empty path connecting C to itself. Also, if (C, C') is a leaking pair, then for any C_1, C_2 that are equivalent to C, C' respectively, the pair (C_1, C_2) is also a leaking pair.

The presence of a leaking pair is also a witness to a DiPA not being differentially private. Consider a DiPA $\mathcal{A}$ that has no leaking cycle but has a leaking pair of cycles (C, C') such that C is reachable. Assume that C' is a G-cycle. The case when C' is an L-cycle is symmetric. Since $\mathcal{A}$ has no leaking cycles, the value stored in x does not change while the automaton is executing the transitions in either C or C'. Let y be the value of x when C' starts executing. One can show that if $y > 0$, then there are a pair of adjacent inputs such that traversing C' on those inputs results in paths such that the ratio of their probabilities is at least $e^{k\epsilon}$ for some k. Moreover, this pair of inputs does not depend on the actual value of y. Further, on these pair of inputs, if $y \leq 0$, then the ratio of these probabilities is ≥ 1. This once again means that by repeating C' ℓ times, we can get adjacent inputs whose probabilities violate the $d\epsilon$ privacy budget (for any d) if $y > 0$. A similar observation holds for L-cycle C — if the value of x at the start of C is ≤ 0, then we can find adjacent inputs such that traversing C for those inputs results in paths whose probabilities have a "high" ratio. Further, on these pair

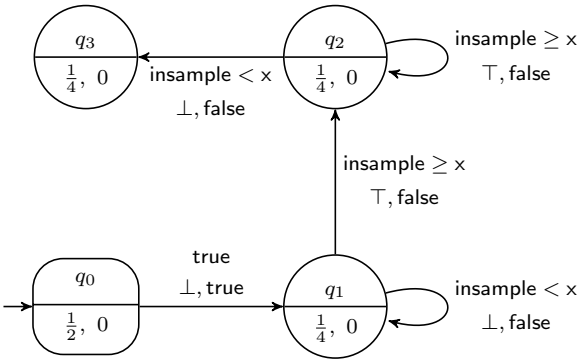

Fig. 4. DiPA $\mathcal{A}_{\mathsf{SVT}*}$ modeling an algorithm that processes a sequence of real numbers and implements a "noisy' version" of the following process. As long as the input numbers are less than threshold T ($= 0$) it outputs $\bot$. Once it sees the first number $\geq T$, it moves to the second phase. In the phase, it outputs $\top$ as long as the numbers are $\geq T$. When it sees the first number $< T$, it outputs $\bot$ and stops. Since $\mathsf{insample}'$ is never output, parameters used in its sampling are not shown and not important.

of inputs, if the value stored in x is > 0, then the ratio of these probabilities is ≥ 1. The next observation is that value stored in x at the end of an AG-path is at least the value at the beginning of the path. We can now put all these pieces together to get our witness for a violation of differential privacy. If the value of x is ≤ 0 at the start of C, then repeating C ℓ times gives us a pair of adjacent inputs that violate the privacy budget. On the other hand, if x at the start of C is > 0, it will be > 0 even at the start of C', and then repeating C' ℓ times gives us the witnessing pair. Let us illustrate the intuition through an example.

Example 5. Consider the automaton $\mathcal{A}_{\mathsf{SVT}*}$ shown in Fig. 4. It implements an algorithm that is a slight modification of Algorithm 1 (or the DiPA $\mathcal{A}_{\mathsf{SVT}}$ in Fig. 1). Like in SVT, the automaton starts in state q_0 by sampling a value that is a perturbed value of a threshold T (which is 0 here). It stores this sampled value in x and moves to the first phase (state q_1). In this phase, the automaton outputs $\bot$ and stays in q_1 as long as a perturbed value of the input read is less than the perturbed threshold stored in x. The first time it encounters a perturbed value that is at least x, it moves to phase two (state q_2) and outputs $\top$. In state q_2, it outputs $\top$ as long as the perturbed inputs it samples are $\geq$ x. The first time it encounters a value $<$ x it outputs $\bot$ and terminates. Throughout the computation, the automaton never over-writes the value stored in the first step in variable x.

$\mathcal{A}_{\mathsf{SVT}*}$ has a leaking pair. Observe that $C = q_1 \xrightarrow{a_1,\bot} q_1$ is an L-cycle and $C' = q_2 \xrightarrow{a_2,\top} q_2$ is a G-cycle. The path $q_1 \xrightarrow{a_3,\top} q_2$ is an AG-path from C to C'. Hence (C, C') is a leaking pair. The presence of this leaking pair can be exploited to show that $\mathcal{A}_{\mathsf{SVT}*}$ is not $d\epsilon$-differentially private for any $d > 0$.

Consider the following two paths.

$$\rho_1^\ell = q_0 \xrightarrow{\tau,\bot} \left[q_1 \xrightarrow{-\frac{1}{2},\bot} q_1\right]^\ell \xrightarrow{0,\top} \left[q_2 \xrightarrow{\frac{1}{2},\top} q_2\right]^\ell \xrightarrow{0,\bot} q_3$$

$$\rho_2^\ell = q_0 \xrightarrow{\tau,\bot} \left[q_1 \xrightarrow{\frac{1}{2},\bot} q_1\right]^\ell \xrightarrow{0,\top} \left[q_2 \xrightarrow{-\frac{1}{2},\top} q_2\right]^\ell \xrightarrow{0,\bot} q_3$$

In the above $[p \xrightarrow{a,o} q]^\ell$ means that the path consists of repeating this transition ℓ times. Notice that the $\mathsf{inseq}(\rho_1^\ell) = (-\frac{1}{2})^\ell 0(\frac{1}{2})^\ell 0$ and $\mathsf{inseq}(\rho_2^\ell) = (\frac{1}{2})^\ell 0(-\frac{1}{2})^\ell 0$ are adjacent. Moreover, for any $d > 0$, there is a ℓ such that for every ϵ the ratio of $\mathsf{Pr}[\epsilon, \rho_1^\ell]$ and $\mathsf{Pr}[\epsilon, \rho_2^\epsilon]$ is $> e^{d\epsilon}$. Thus, for an appropriately chosen value for ℓ, ρ_1^ℓ and ρ_2^ℓ witness the violation of differential privacy.

The two conditions we have identified thus far — existence of reachable leaking cycle or leaking pair — demonstrate differential privacy violations even in DiPAs that do not output any real value. In automata that output real values, there are additional sources of privacy violations. We identify these conditions next.

Definition 9. A cycle C of a DiPA $\mathcal{A}$ is a *disclosing cycle* if there is an i, $0 \leq i < |C|$ such that $\mathsf{trans}(C[i])$ is an input transition that outputs either $\mathsf{insample}$ or $\mathsf{insample}'$.

Again the existence of a reachable disclosing cycle demonstrates that the DiPA is not differentially private — outputting a perturbed input repeatedly exhausts the privacy budget.

We now present the last property of importance that pertains to paths that have transitions that output the value of insample. We say that a state q *is in a cycle* (G-cycle or L-cycle) if there is a cycle (G-cycle/L-cycle) C and index i such that $q = \mathsf{state}(C[i])$.

Definition 10. We say that a path $\rho = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$ of length n of DiPA $\mathcal{A}$ is a *privacy violating path* if one of the following conditions hold.

- $\mathsf{tail}(\rho)$ is an AG-path (resp., AL-path) such that $\mathsf{last}(\rho)$ is in a G-cycle (resp., L-cycle) and the 0th transition $\mathsf{trans}(\rho[0])$ is an assignment transition that outputs insample.
- ρ is an AG-path (resp., AL-path) such that $\mathsf{last}(\rho)$ is in a G-cycle (resp., L-cycle) and the 0th transition has $\mathsf{guard}(\rho[0]) = \mathsf{insample} < \mathsf{x}$ (resp., $\mathsf{guard}(\rho[0]) = \mathsf{insample} \geq \mathsf{x}$) and outputs insample.
- ρ is an AG-path (resp., AL-path) such that $\mathsf{first}(\rho)$ is in an L-cycle (resp., G-cycle) and the last transition has guard $\mathsf{guard}(\rho[n-1]) = \mathsf{insample} \geq \mathsf{x}$ (resp., $\mathsf{guard}(\rho[n-1]) = \mathsf{insample} < \mathsf{x}$) and outputs insample.

Once again, the presence of a reachable privacy violating path demonstrates that the automaton is not differentially private. Let us provide some intuition why that is the case. We do this for some of the cases that form a privacy violating path with reasoning for the missing cases being similar. As before, let us assume that there is no leaking cycle because if there is one then we already know that the automaton is not differential privacy. A consequence of this that there are no

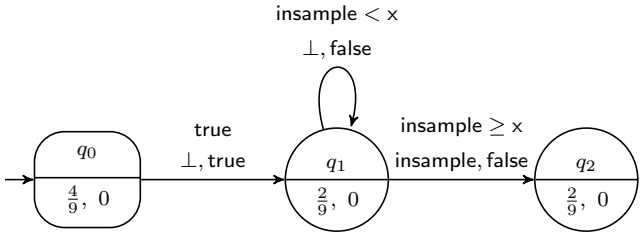

Fig. 5. DiPA $\mathcal{A}_{\mathsf{mod}}$ is a modification of $\mathcal{A}_{\mathsf{NumSp}}$. Label of each state below the line shows the parameters for sampling insample. Parameters for sampling insample$'$ are not shown in the figure; they are $\frac{1}{9}$ (scaling factor) and 0 (mean) in every state.

assignment transitions in a G-cycle or L-cycle and hence the value stored in x remains unchanged in these cycles. Let us recall a couple of crucial observations that we used when we argued in the case of a leaking pair. First, the value stored in x at the end of an AG-path is at least as large as the value at the beginning. Next, if a G-cycle (L-cycle) is traversed when the starting value in x is > 0 (≤ 0) then we have a family of pairs of adjacent inputs that correspond to traversing the cycle multiple times with the property that the ratio of their probabilities diverges as the cycle is traversed more times. Let us now consider each of the cases in the definition of privacy violating path. If ρ starts with an assignment transition that outputs insample and if the output of this first step is in the interval $(0, \infty)$ then the value of x is > 0 at the end of ρ when a G-cycle can be traversed. These observations can be used to give us a pair of adjacent inputs that violate privacy. If ρ starts with a transition whose guard is insample $<$ x that outputs insample and suppose the value output in this step is in the interval $(0, \infty)$ then the value in x at the start is > 0. Like in the previous case this can be used to get a violating pair of inputs. Finally, if ρ ends in transition outputting insample, guard insample $\geq$ x and the value output in this last step in the interval $(-\infty, 0)$, then we can conclude that the value in x at the end of ρ is ≤ 0. This combined with properties of AG-paths means that x has a value ≤ 0 at the beginning of ρ. This means the L-cycle at the start of ρ can be traversed with x having a value ≤ 0 which means that a violating pair of inputs can be constructed.

Let us illustrate this last condition through another example.

Example 6. Consider automaton $\mathcal{A}_{\mathsf{mod}}$ (Fig. 5) which is a modification of the Numeric Sparse algorithm modeled by automaton $\mathcal{A}_{\mathsf{NumSp}}$ (Fig. 2). The only difference is that the transition from q_1 to q_2 outputs insample as opposed to insample$'$. This change causes this automaton to be not differentially private.

Observe that the state q_1 is in a L-cycle $q_1 \xrightarrow{a,\perp} q_1$ and then path $\rho = q_1 \xrightarrow{a,(\mathsf{insample},(0,\infty))} q_2$ is an AG-path. Finally, the last transition (or rather the only transition) of ρ has guard insample $\geq$ x that outputs insample. Thus, ρ is a privacy violating path.

We can use ρ to find a violation for privacy. Consider the following pair of paths.

$$\rho_1^\ell = q_0 \xrightarrow{\tau,\perp} \left[q_1 \xrightarrow{-\frac{1}{2},\perp} q_1\right]^\ell \xrightarrow{0,(\mathsf{insample},(0,\infty))} q_2$$

$$\rho_2^\ell = q_0 \xrightarrow{\tau,\perp} \left[q_1 \xrightarrow{\frac{1}{2},\perp} q_1\right]^\ell \xrightarrow{0,(\mathsf{insample},(0,\infty))} q_2$$

Observe that $\mathsf{inseq}(\rho_1^\ell) = (-\frac{1}{2})^\ell 0$ and $\mathsf{inseq}(\rho_2^\ell) = (\frac{1}{2})^\ell 0$ are adjacent. Moreover, for any $d > 0$, there is an ℓ such that for any ϵ, the ratio of $\mathsf{Pr}[\epsilon, \rho_1^\ell]$ and $\mathsf{Pr}[\epsilon, \rho_2^\ell]$ is $> e^{d\epsilon}$. Thus, ρ_1^ℓ and ρ_2^ℓ demonstrate the violation of privacy.

As the discussion and examples above illustrate, absence of leaking cycles, leaking pairs, disclosing cycles, and privacy violating paths is necessary for a DiPA to be differentially private. We call such automata *well-formed*.

Definition 11. A DiPA $\mathcal{A}$ is said to be *well-formed* if $\mathcal{A}$ has no reachable leaking cycle, no leaking pair (C, C') where C is reachable, no reachable disclosing cycle, and no reachable privacy violating path.

Our main theorem is that well-formed DiPAs are exactly the class of automata that are differentially private. The proof of this Theorem is carried out in the Appendix (See Appendix B for the "only if" direction and Appendix C for the "if" direction).

Theorem 2. *Let $\mathcal{A}$ be a DiPA. There is a $d > 0$ such that for every $\epsilon > 0$, $\mathcal{A}$ is $d\epsilon$-differentially private if and only if $\mathcal{A}$ is well-formed.*

Remark. Before presenting a proof sketch for Theorem 2, it is useful to point out one special case for the result. Observe that disclosing cycles and privacy violating paths pertain to paths that have transitions that output real values. For DiPAs that do not have real outputs, disclosing cycles and privacy violating paths are not needed to get an exact characterization of differential privacy. More precisely, we say that a DiPA $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$ has *finite valued outputs* if every transition in $\mathcal{A}$ outputs a value in Γ. Now, a DiPA with finite valued outputs is differentially private if and only if it has no reachable leaking cycles and leaking pairs.

Discussion in this section has provided intuitions for why well-formed-ness is necessary for an automaton to be differentially private; the formal proof that captures these intuitions is subtle, long, and non-trivial. The proof is postponed to Appendix B. We sketch some key properties that show why it is sufficient.

Let us fix a transition $t = (p, c, q, o, b)$ in a DiPA $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$. The transition t is said to lie on a cycle if there is a reachable cycle ρ and index i such that $\mathsf{trans}(\rho[i]) = t$. On the other hand, we will say t is a *critical transition* if t does not lie on a cycle. Let $P(p) = (d, \mu, d', \mu')$ be the parameters for sampling insample and insample$'$ in state

p. We define the cost of t as follows.

$$\mathsf{cost}(t) = \begin{cases} d & t \text{ is a critical non-input transition} \\ 2d & t \text{ is a critical input transition and} \\ & o \neq \mathsf{insample}' \\ 2d + d' & t \text{ is a critical input transition and} \\ & o = \mathsf{insample}' \\ 0 & \text{otherwise} \end{cases}.$$

For a path ρ, define weight of ρ as $\mathsf{wt}(\rho) = \sum_{i=0}^{|\rho|-1} \mathsf{cost}(\mathsf{trans}(\rho[i]))$, i.e., the sum of the costs of all the transitions in ρ. Finally, define $\mathsf{wt}(\mathcal{A})$ to be the supremum over all paths ρ, $\mathsf{wt}(\rho)$. In fact, the weight of $\mathcal{A}$ could have been defined as a maximum (as opposed to a supremum) because they are the same in this case. The crucial observation about weight of an automaton that is used in proving the sufficiency of well-formed-ness for differential privacy, is that it provides an upper bound on the privacy budget for $\mathcal{A}$.

Lemma 3. *A well-formed DiPA $\mathcal{A}$ is $\mathsf{wt}(\mathcal{A})\epsilon$-differentially private for all $\epsilon > 0$.*

Proof. (Sketch.) The Lemma is a consequence of the proof of Lemma 13 given in Appendix C. This lemma relates the probabilities of two paths ρ and ρ' of $\mathcal{A}$, such that ρ and ρ' start from the same reachable state, ρ and ρ' are equivalent, $\mathsf{inseq}(\rho)$ and $\mathsf{inseq}(\rho')$ are neighbors, and the initial transitions of ρ and ρ' are assignment transitions. More precisely, for an initial value x_0 of x, Lemma 13 shows that $\mathsf{Pr}[\epsilon, x_0, \rho']$ is at least $e^{-\mathsf{wt}(\rho)\epsilon}$ times one of three quantities: $\mathsf{Pr}[\epsilon, x_0, \rho]$, $\mathsf{Pr}[\epsilon, x_0 + 1, \rho]$ or $\mathsf{Pr}[\epsilon, x_0 - 1, \rho]$. The specific quantity the Lemma compares $\mathsf{Pr}[\epsilon, x_0, \rho']$ to depends on some properties of the path ρ stated in Lemma 13. Together these mutually exclusive properties serve as an exhaustive list of properties that the path ρ can satisfy. The fact that the list is exhaustive is a consequence of well-formed-ness. In particular, one of the parts of the Lemma is that when the guard of the initial transition is true then $\mathsf{Pr}[\epsilon, x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[\epsilon, x_0, \rho]$. This immediately implies the statement of the current Lemma. The proof of Lemma 13 itself is intricate and proceeds by induction on the number of assignment transitions in ρ. □

Example 7. Let us consider the automata $\mathcal{A}_{\mathsf{SVT}}$ (Fig. 1 on Page 5) and $\mathcal{A}_{\mathsf{NumSp}}$ (Fig. 2 on Page 5). Both these automata are well-formed and hence they are differentially private. Moreover, we can use Lemma 3 to provide an upper bound on the required privacy budget.

Observe that the only critical transitions in $\mathcal{A}_{\mathsf{SVT}}$ are t_{01}, the transition from q_0 to q_1, and t_{12}, the transition from q_1 to q_2. Now $\mathsf{cost}(t_{01}) = \frac{1}{2}$, while $\mathsf{cost}(t_{12}) = 2(\frac{1}{4}) = \frac{1}{2}$. Thus, $\mathsf{wt}(\mathcal{A}_{\mathsf{SVT}}) = \frac{1}{2} + \frac{1}{2} = 1$, or $\mathcal{A}_{\mathsf{SVT}}$ is ϵ-differentially private for all ϵ.

Similarly, the only critical transitions in $\mathcal{A}_{\mathsf{NumSp}}$ are again transition t_{01} from q_0 to q_1 and transition t_{12} from q_1 to q_2. They have the following costs: $\mathsf{cost}(t_{01}) = \frac{4}{9}$ and $\mathsf{cost}(t_{12}) = 2(\frac{2}{9}) + \frac{1}{9} = \frac{5}{9}$. Thus, $\mathsf{wt}(\mathcal{A}_{\mathsf{NumSp}}) = \frac{4}{9} + \frac{5}{9} = 1$ and $\mathcal{A}_{\mathsf{NumSp}}$ is ϵ-differentially private for all $\epsilon > 0$.

In both cases, the upper bounds computed through our methods match the known upper bounds.

Remark. Observe that the means used in sampling $\mathsf{insample}$ and $\mathsf{insample}'$ do not play any role in the definition of well-formed (Definition 11). They also do not play any role in the calculation of the weight of an automaton or Lemma 3. This allows one to make some simple observations. Recall that $\mathcal{A}_{\mathsf{SVT}}$ and $\mathcal{A}_{\mathsf{NumSp}}$ were defined by taking the threshold $T = 0$. However, these observations allow us to conclude that no matter what value is chosen for the threshold T, $\mathcal{A}_{\mathsf{SVT}}$ and $\mathcal{A}_{\mathsf{NumSp}}$ are ϵ-differentially private for all $\epsilon > 0$.

We get as a corollary of Theorem 2 that the problem of checking whether a DiPA $\mathcal{A}$ is differentially private can be checked using graph-theoretic algorithms in linear time.

Corollary 4. *The differential privacy problem for DiP automata is decidable in linear time. In addition, $\mathsf{wt}(\mathcal{A})$ can be computed in linear time, assuming addition and comparison of numbers takes constant time.*

Proof. We describe a linear time algorithm that checks whether a DiPA $\mathcal{A}$ is well-formed. The Corollary then follows from Theorem 2.

Let us fix $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$. Consider the edge-labeled directed graph $\mathcal{G}$ whose vertex set is Q and there is an edge-labeled (c, b) from p to q if $\delta(p, c) = (q, o, b)$ for some o. Without loss of generality, we can assume that every state is reachable from q_{init}. It is worth observing that because of the determinism condition of DiPAs, the number of edges in $\mathcal{G}$ is at most twice the number of vertices. The subgraph $\mathcal{G}_{\mathsf{AG}}$ of $\mathcal{G}$ has the same vertex set but an edge labeled (c, b) is present in $\mathcal{G}_{\mathsf{AG}}$ only if whenever $b = \mathsf{true}$, $c = \mathsf{insample} \geq \mathsf{x}$. Similarly, the subgraph $\mathcal{G}_{\mathsf{AL}}$ of $\mathcal{G}$ only has those edges labeled (c, b) with the property that if $b = \mathsf{true}$ then $c = \mathsf{insample} < \mathsf{x}$. Notice that the graphs $\mathcal{G}$, $\mathcal{G}_{\mathsf{AG}}$ and $\mathcal{G}_{\mathsf{AL}}$ can each be constructed in linear time from $\mathcal{A}$.

Next, we compute the maximal strongly connected components (SCC) of $\mathcal{G}$; this can also be done in linear time. Observe that a state q is part of some G-cycle if it's SCC has an edge with label $(\mathsf{insample} \geq \mathsf{x}, b)$. Similarly, q is part of some L-cycle if it's SCC has an edge with label $(\mathsf{insample} < \mathsf{x}, b)$. Notice that the set of all states that belong to some G-cycle and those that belong to some L-cycle can be computed in linear time. Next, the set of all vertices that can be reached by an AG-path from an L-cycle can be computed in linear time by performing a BFS on $\mathcal{G}_{\mathsf{AG}}$ starting from vertices that are on L-cycles. Similarly, we can compute all vertices from which a G-cycle can be reached by an AG-path in linear time. Using BFS on $\mathcal{G}_{\mathsf{AL}}$ we can also compute the set of all vertices that can be reached from a G-cycle by an AL-path, and the set of all vertices from which an L-cycle can be reached by an AL-path in linear time.

We can now check each of the conditions of well-formed-ness in linear time using the sets computed in the previous

paragraph.

- *leaking cycle*: Check if there is a SCC of $\mathcal{G}$ that has an edge labeled (c, true) and an edge labeled (c', b') where $c' \neq \mathsf{true}$.
- *leaking pair*: Check if there is a state on an L-cycle that can reach a G-cycle by an AG-path and check if there is a state on an G-cycle that can reach a L-cycle by an AL-path.
- *disclosing cycle*: Check if there is a SCC of $\mathcal{G}$ that contains an edge from an input state that outputs $\mathsf{insample}$ or $\mathsf{insample}'$.
- *privacy violating path*: Check if any of the following conditions holds: (a) there is an AG-path (AL-path) from the target of an assignment transition to a state on a G-cycle (L-cycle); (b) there is an AG-path (AL-path) from the target of a non-assignment transition with output $\mathsf{insample}$ and guard $\mathsf{insample} < \mathsf{x}$ ($\mathsf{insample} \geq \mathsf{x}$) to a state on a G-cycle (L-cycle); (c) there is an AG-path (AL-path) from a state on an L-cycle (G-cycle) to the source of a transition with guard $\mathsf{insample} \geq \mathsf{x}$ ($\mathsf{insample} < \mathsf{x}$) that outputs $\mathsf{insample}$.

We now show how $\mathsf{wt}(\mathcal{A})$ can be computed in linear time assuming that arithmetic operations take constant time. Observe that we can construct the graph of SCCs of $\mathcal{G}$ in linear time and that critical transitions are those that correspond to edges in this graph of SCCs. $\mathsf{wt}(\mathcal{A})$ is the length of the longest path in this graph, where the weight of an edge is the cost of the corresponding transition. Note that this can be computed in linear time because the graph of SCCs is a DAG. □

Remark. Observe that the well-formed-ness of an automata $\mathcal{A}$ does not depend on the parameter function P of the automata. Hence, once we have established that $\mathcal{A}$ is differentially private, we establish it for all possible parameter functions. The weight of a well-formed $\mathcal{A}$, however, does indeed with the scaling parameters given by P. It is independent of the mean parameters given by P.

V. Related Work

Privacy proof construction: Several works [6], [7], [11], [12], [15], [20] have proposed the use of type systems to construct proofs of differential privacy. Some of the type-based approaches such as [6], [7], [12], [20] rely on linear dependent types, for which the type-checking and type-inference may be challenging. For example, the type checking problem for the type system in [20] is undecidable. The type systems in Zhang and Kifer [11], later expanded on in [15], rely on using the techniques of randomness alignments and can handle advanced examples such as the sparse vector technique. Barthe et al. [8]–[10] develop several program logics based on probabilistic couplings for reasoning about differential privacy, which have been used successfully to analyze standard examples from the literature, including the sparse vector technique. The probabilistic couplings and randomness alignment arguments are synthesized into coupling strategies by Albarghouthi and Hsu [13]. A shadow execution based method is introduced in [14]. Both [13] and [14] are automated and can handle advanced examples such as sparse vector technique efficiently. M. C. Tschantz et. al use probabilistic I/O automata in [21] to model interactive differential privacy algorithms. Simulation-based methods are used to verify differential privacy. They assume that inputs and outputs take values from a discrete domain and that the sampling is from discrete probability distributions. While these approaches can handle arbitrarily long sequences of inputs and verify ϵ-differential privacy, they are not shown to be complete and may fail to construct a proof of differential privacy even when the mechanism is differentially private.

Counterexample generation: Another investigation line develops automated techniques to search for privacy violations. Ding et al. [16] use statistical techniques based on hypothesis testing for automatic generation of counterexamples. Bischel et al. [17] use optimization-based techniques and symbolic differentiation to search for counterexamples. These methods search only amongst a bounded sequence of inputs and assume a concrete value of the parameter ϵ. Wang et al. [15] use program analysis techniques to generate counterexamples when it fails to construct a proof.

Model-checking/Markov Chain approaches: The probabilistic model checking approach for verifying ϵ-differential privacy is employed in [22], [23], where it is assumed that the program is given as a Markov Chain. These approaches do not allow for sampling from continuous random variables. Instead, they assume that the program behavior is given as a finite Markov Chain, and the transition probabilities are specified as inputs. Thus, they also implicitly assume a bounded sequence of inputs and a concrete value of ϵ. In [24], the authors use labeled Markov Chains to model differential privacy algorithms. They consider discrete probabilities only. They only model inputs taking values from a finite set and implicitly assume a concrete value of ϵ. Further, they check whether the ratio of probabilities of observations on neighboring inputs is bounded by a constant. If it is bounded, it implies the algorithm is ϵ-differentially private for sufficiently large ϵ.

Decision Procedures: The decision problem of checking whether a randomized program is differentially private is studied in [5], where it is shown to be undecidable for programs with a single input and single output, assuming that the program can sample from Laplacian distributions. They identify a language that restricts the mechanisms in order to obtain decidability. The restriction forces sampling from the Laplace distribution only a bounded number of times. The number of inputs and outputs are also bounded and constrained to take values from a finite domain. The decision procedure in [5] relies on the decision procedure for checking the validity of a sentence in the fragment of the theory of Reals with exponentiation identified in [18], and has very high complexity. The decision procedure allows for verification of differential privacy for all ϵ.

Complexity: Gaboardi et. al [25] study the complexity of deciding differential privacy for randomized Boolean circuits, and show that the problem is $\mathbf{coNP}^{\#\mathbf{P}}$-complete. They

assume finite number of inputs, the only probabilistic choices in [25] are fair coin tosses, and e^ϵ is taken to be a fixed rational number.

VI. Conclusion

In this paper, we introduced a model called DiP automata for modeling differential privacy mechanisms. Such automata can be used to model some of the interesting classes of mechanisms presented in the literature. We studied the problem of checking if a mechanism given by a DiPA is *differentially private*, i.e., it is $d\epsilon$-differentially private, for some constant $d > 0$ and for all values of the scaling parameter $\epsilon > 0$. We showed that this problem is decidable in time that is linear in the size of the automaton. Our decidability result is based on checking the necessary and sufficient conditions for differential privacy, presented in the paper. If the mechanism, given by an automaton, is differentially private, then it outputs a constant d such that the mechanism is $d\epsilon$-differentially private, for all $\epsilon > 0$. If the mechanism is not differentially private, a counterexample can be constructed explaining why it is not differentially private. For the published mechanisms presented in the literature, that are differentially private, the constant d computed by our method matches the published values. The proofs showing that the given conditions presented in the paper, are necessary and sufficient for differential privacy, are highly non-trivial.

As part of future work, it will be interesting to come up with the computation of the optimal constant d for mechanisms modeled by DiPA, that are differentially private. It will be interesting to extend our automata model to cover other interesting differential privacy mechanisms such as private smart sum algorithm [26], private vertex cover [27] and NoisyMax [2].

Acknowledgment

The authors would like to thank anonymous reviewers for their interesting and valuable comments. Rohit Chadha was partially supported by NSF CNS 1553548 and NSF CCF 1900924. A. Prasad Sistla was partially supported by NSF CCF 1901069, and Mahesh Viswanathan was partially supported by NSF NSF CCF 1901069 and NSF CCF 2007428.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *IACR Theory of Cryptography Conference (TCC)*, 2006, pp. 265–284.

[2] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[3] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. P. Vadhan, "On the complexity of differentially private data release: efficient algorithms and hardness results," in *ACM SIGACT Symposium on Theory of Computing (STOC)*, 2009, pp. 381–390.

[4] M. Lyu, D. Su, and N. Li, "Understanding the sparse vector technique for differential privacy," *Proceedings of VLDB*, vol. 10, no. 6, pp. 637–648, 2017.

[5] G. Barthe, R. Chadha, V. Jagannath, A. P. Sistla, and M. Viswanathan, "Deciding differential privacy for programs with finite inputs and outputs," in *35th Annual ACM/IEEE Symposium on Logic in Computer Science*, 2020, pp. 141–154.

[6] J. Reed and B. C. Pierce, "Distance makes the types grow stronger: A calculus for differential privacy," in *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, 2010, p. 157–168.

[7] M. Gaboardi, A. Haeberlen, J. Hsu, A. Narayan, and B. C. Pierce, "Linear dependent types for differential privacy," in *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL)*, 2013, pp. 357–370.

[8] G. Barthe, B. Köpf, F. Olmedo, and S. Zanella-Béguelin, "Probabilistic relational reasoning for differential privacy," *ACM Transactions on Programming Languages and Systems*, vol. 35, no. 3, p. 9, 2013.

[9] G. Barthe, M. Gaboardi, B. Grégoire, J. Hsu, and P.-Y. Strub, "Proving differential privacy via probabilistic couplings," in *IEEE Symposium on Logic in Computer Science (LICS)*, 2016.

[10] G. Barthe, N. Fong, M. Gaboardi, B. Grégoire, J. Hsu, and P. Strub, "Advanced probabilistic couplings for differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 55–67.

[11] D. Zhang and D. Kifer, "LightDP: towards automating differential privacy proofs," in *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 2017, pp. 888–901.

[12] A. A. de Amorim, M. Gaboardi, J. Hsu, and S. Katsumata, "Probabilistic relational reasoning via metrics," in *34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2019, pp. 1–19.

[13] A. Albarghouti and J. Hsu, "Synthesizing coupling proofs of differential privacy," in *Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 2018, pp. 58:1–58:30.

[14] Y. Wang, Z. Ding, G. Wang, D. Kifer, and D. Zhang, "Proving differential privacy with shadow execution," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, (PLDI)*, 2019, pp. 655–669.

[15] Y. Wang, Z. Ding, D. Kifer, and D. Zhang, "CheckDP: An automated and integrated approach for proving differential privacy or finding precise counterexamples," in *2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020, pp. 919–938.

[16] Z. Ding, Y. Wang, G. Wang, D. Zhang, and D. Kifer, "Detecting violations of differential privacy," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018, pp. 475–489.

[17] B. Bichsel, T. Gehr, D. Drachsler-Cohen, P. Tsankov, and M. T. Vechev, "DP-Finder: Finding differential privacy violations by sampling and optimization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018, pp. 508–524.

[18] S. McCallum and V. Weispfenning, "Deciding polynomial-transcendental problems," *Journal of Symbolic Computation*, vol. 47, no. 1, pp. 16–31, 2012.

[19] R. Chadha, A. P. Sistla, and M. Viswanathan, "On linear time decidability of differential privacy for programs with unbounded inputs," in *36th Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2021, To Appear.

[20] A. A. de Amorim, M. Gaboardi, E. J. G. Arias, and J. Hsu, "Really natural linear indexed type checking," in *26th 2014 International Symposium on Implementation and Application of Functional Languages (IFL)*, 2014, pp. 5:1–5:12.

[21] M. C. Tschantz, D. K. Kaynar, and A. Datta, "Formal verification of differential privacy for interactive systems (extended abstract)," in *27th Conference on the Mathematical Foundations of Programming Semantics (MFPS)*, ser. Electronic Notes in Theoretical Computer Science, vol. 276, 2011, pp. 61–79.

[22] K. Chatzikokolakis, D. Gebler, C. Palamidessi, and L. Xu, "Generalized bisimulation metrics," in *35th International Conference on Concurrency Theory (CONCUR)*, 2014, pp. 32–46.

[23] D. Liu, B. Wang, and L. Zhang, "Model checking differentially private properties," in *Programming Languages and Systems - 16th Asian Symposium, (APLAS)*, ser. Lecture Notes in Computer Science, vol. 11275, 2018, pp. 394–414.

[24] D. Chistikov, S. Kiefer, A. S. Murawski, and D. Purser, "The big-o problem for labelled markov chains and weighted automata," in *31st International Conference on Concurrency Theory (CONCUR)*, ser. LIPIcs, vol. 171, 2020, pp. 41:1–41:19.

[25] M. Gaboardi, K. Nissim, and D. Purser, "The complexity of verifying loop-free programs as differentially private," in *47th International Colloquium on Automata, Languages, and Programming, (ICALP)*, ser. LIPIcs, vol. 168, 2020, pp. 129:1–129:17.

[26] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security*, vol. 14, no. 3, p. 26, 2011.

[27] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar, "Differentially private combinatorial optimization," in *ACM–SIAM Symposium on Discrete Algorithms (SODA)*, 2010, pp. 1106–1125.

APPENDIX A
AUXILIARY DEFINITIONS

We shall start by defining some auxiliary definitions that shall help us in the proof of Theorem 2.

Path Suffixes: Let $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$ be an DiP automaton. For any execution/path $\eta = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$ of $\mathcal{A}$ and $i < n$, the suffix of ρ starting from state q_i (or position i) is the path $q_i \xrightarrow{a_i,o_i} q_{i+1} \xrightarrow{a_{i+1},o_{i+1}} q_{i+2} \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$ and is denoted as $\rho||i$.

Abstract paths: For any execution/path $\eta = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$ of $\mathcal{A}$, the *abstraction* of ρ, denoted $\mathsf{abstract}(\rho)$, will be the word $q_0\sigma_0q_1\sigma_1 \cdots q_{n-1}\sigma_{n-1}q_n$ where

$$\sigma_i = \begin{cases} o_i & \text{if } o_i \in \Gamma \\ \mathsf{insample} & \text{if } o_i = (\mathsf{insample}, r, s) \\ \mathsf{insample}' & \text{otherwise} \end{cases}$$

Note that for DiP automata, $\sigma_i = o_i$ for each i.

A sequence $\eta = q_0\sigma_0q_1\sigma_1 \cdots q_{n-1}\sigma_{n-1}q_n$ is said to be an *abstract path* if $\eta = \mathsf{abstract}(\rho)$ for some execution ρ. By abuse of notation, we shall say that the length pf the execution η is n. Further such a ρ shall be called an execution of η on input $\alpha = a_0 \cdots a_n$. Note that ρ is unique if $\sigma_i \in \Gamma$ for each i. In general, two distinct sequences ρ and ρ' having the same abstraction η will only differ at indices i such that $\sigma_i \notin \Gamma$. At those indices, we would need to specify the values of the interval end-points, r_i, s_i, where the real output is assumed to belong to.

Fix an abstract path $\eta = q_0\sigma_0q_1\sigma_1 \cdots q_{n-1}\sigma_{n-1}q_n$. The ith-transition, denoted $\mathsf{trans}[i]$, is the word $q_i\sigma_iq_{i+1}$. The guard of the ith transition, denoted $\mathsf{guardtrans}[i]$ is the unique c such that $\delta(q_i, c) = (q_i, \sigma_i, b)$. The output sequence of η, denoted $\mathsf{outseq}(\eta)$ is the sequence $\sigma_0 \cdots \sigma_n$. Note that we can classify transitions of an abstract path as input, non-input, assignment and non-assignment as expected. The notions of paths, cycles, reachability, leaking cycle, leaking pair, disclosing cycle, privacy violating path and critical transition extends naturally to abstract paths.

APPENDIX B
NECESSITY OF WELL-FORMEDNESS

We shall now show that if the DiPA $\mathcal{A}$ is not well-formed then $\mathcal{A}$ is not differentially private, thus establishing the "only if" part of Theorem 2. The proof of necessity will be broken into four Lemmas. Lemma 6 shall show that if $\mathcal{A}$ has a leaking cycle then $\mathcal{A}$ is not differentially private. Lemma 7 will deal with presence of leaking pairs, Lemma 8 with presence of disclosing cycles, and Lemma 9 with presence of privacy violating paths. Please note that we shall use the notions of path suffixes and abstract paths introduced in Appendix A.

Before we proceed, we need a technical lemma that characterizes the probability of two samples from Laplace distributions being ordered.

Lemma 5. *Suppose X_i, for $i = 1, 2$, are random variables with $X_i \sim \mathsf{Lap}(k_i, \mu_i)$. Then $\mathsf{Prob}[X_1 \leq X_2]$ is given as follows. When $k_1 \neq k_2$*

$$\mathsf{Prob}[X_1 \leq X_2] = \frac{1}{2}\left[1 + \mathsf{sgn}(\mu_2 - \mu_1)\left(1 - \frac{k_2^2}{2(k_2^2 - k_1^2)}e^{-k_1|\mu_2 - \mu_1|} + \frac{k_1^2}{2(k_2^2 - k_1^2)}e^{-k_2|\mu_2 - \mu_1|}\right)\right].$$

On the other hand, when $k_1 = k_2 = k$

$$\mathsf{Prob}[X_1 \leq X_2] = \frac{1}{2}\left[1 + \mathsf{sgn}(\mu_2 - \mu_1)\left(1 - e^{-k|\mu_2 - \mu_1|}(1 + \frac{k}{2} \mid \mu_2 - \mu_1|)\right)\right]$$

Leaking cycles implies no privacy

Lemma 6. *A DiPA $\mathcal{A}$ is not differentially private if it has a reachable leaking cycle.*

Let $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$. Assume that $\mathcal{A}$ has a leaking cycle reachable from the state q_{init}. We give the proof first assuming that all states of $\mathcal{A}$ are input states. The proof for the case when the automata has both input and non-input states can be proved along similar lines and is left out.

Let $\eta = q_0\sigma_0q_1\sigma_1 \cdots q_{m+n-1}\sigma_{m+n-1}q_{m+n}$ for $k = 0, \ldots, m + n - 1$ be an abstract path such that $q_0 = q_{\mathsf{init}}$, $q_m = q_{m+n}$, and the final n transitions of ρ, i.e., the abstract path $C = q_m\sigma_mq_{m+1}\sigma_{m+1} \cdots q_{m+n-1}\sigma_{m+n-1}q_{m+n}$ is a leaking cycle.

Let t_k be the k-th transition of η and c_k be the guard of the k-th transition. Further, let d_k and μ_k be such that $P(q_k) = (d_k, \mu_k)$ for each k. We have that $c_0 = \mathsf{true}$ and t_0 is an assignment transition. Let i, j be the smallest integers such that $m \leq i < j < m + n$ and the following properties are satisfied: (a) t_i is an assignment transition, (b) $c_j \neq \mathsf{true}$ and (c) for every k_1 such that $i < k_1 < j$, t_{k_1} is a non-assignment transition and $c_{k_1} = \mathsf{true}$. We fix i, j as above. Consider any integer $\ell > 0$. We define an abstract path η_ℓ starting from q_{init} by repeating the cycle $t_m, \ldots t_{m+n-1}$, ℓ times. Formally, $\eta_\ell = q_0\sigma_0q_1\sigma_1 \cdots q_{m+\ell n-1}\sigma_{m+\ell n-1}q_{m+\ell n}$ such that $q_k = q_{k-n}$ and $\sigma_k = \sigma_{k-n}$ for $m + n \leq k \leq m + \ell n$. Let $\gamma(\ell) = o_0 \cdots o_{m+\ell n-1}$ be the output sequence of length $m + \ell n$ such that $o_k = \sigma_k$ if $\sigma_k \in \Gamma$, otherwise $o_k = (\sigma_k, -\infty, \infty)$. Once again, we let t_k be the k-th transition of η_ℓ and c_k be the guard of the k-th transition. Now, given $\ell > 0$, we define two neighboring input sequences $\alpha(\ell) = a_0 \cdots a_{m+\ell n-1}$ and $\beta(\ell) = b_0 \cdots b_{m+\ell n-1}$ each of length $m + \ell n$.

The sequence $\alpha(\ell)$ is chosen so that all the guards in the transitions of η_ℓ are satisfied with joint probability $> \frac{1}{2}$ for

large ϵ. The input $a_0 = 0$ and for $0 < k < m + \ell n$, a_k is defined inductively as given below: let $k' < k$ be the largest integer such that $t_{k'}$ is an assignment transition, then a_k is given as follows: if c_k is the guard $\mathsf{insample} \geq \mathsf{x}$ then $a_k = \mu_{k'} - \mu_k + a_{k'} + 1$, otherwise $a_k = \mu_{k'} - \mu_k + a_{k'} - 1$.

Now, consider any k, $0 \leq k < m + \ell n$, such that $c_k \neq \mathsf{true}$ and fix it. Let $k' < k$ be the largest integer such that $t_{k'}$ is an assignment transition. Let $X_{k'}, X_k$ be the two random variables with distributions given by $\mathsf{Lap}(d_{k'}\epsilon, a_{k'})$ and $\mathsf{Lap}(d_k\epsilon, a_k)$. Let Y_k denote the random variable denoting the k^{th} output of η on the input sequence $\alpha(\ell)$. Now consider the case when c_k is $\mathsf{insample} \geq \mathsf{x}$. From the way, we defined $\alpha(\ell)$ it is the case that $\mu_k + a_k = \mu_{k'} + a_{k'} + 1$. Now $\mathsf{Prob}[Y_k \neq o_k] = \mathsf{Prob}[X_k < X_{k'}] = \mathsf{Prob}[X_k \leq X_{k'}]$. Let $d_{\mathsf{mx}} = \max(d_k, d_{k'})$ and $d_{\mathsf{mn}} = \min(d_k, d_{k'})$. From Lemma 5, we see that if $d_k \neq d_{k'}$ then

$$\mathsf{Prob}[X_k \leq X_{k'}] < \frac{{d_{\mathsf{mx}}}^2}{2({d_{\mathsf{mx}}}^2 - {d_{\mathsf{mn}}}^2)} e^{-d_{\mathsf{mn}}\epsilon}.$$

If $d_k = d_{k'}$ then

$$\mathsf{Prob}[X_k \leq X_{k'}] < \frac{1}{2} e^{-d_k\epsilon}(1 + \frac{d_k\epsilon}{2}).$$

From the above, we see that

$$\mathsf{Prob}[Y_k \neq o_k] \leq r e^{-d_{\mathsf{mn}}\epsilon}(1 + \frac{d_{\mathsf{mx}}\epsilon}{2})$$

where r is a constant that depends only on $\mathcal{A}$ (and not on k). Now consider the case when c_k is $\mathsf{insample} < \mathsf{x}$. In this case, $\mu_k + a_k = \mu_k + a_{k'} - 1$ and $\mathsf{Prob}[Y_k \neq o_k] = \mathsf{Prob}[X_{k'} < X_k]$. By a similar analysis, in this case also,

$$\mathsf{Prob}[Y_k \neq o_k] \leq r e^{-d_{\mathsf{mn}}\epsilon}(1 + \frac{d_{\mathsf{mx}}\epsilon}{2}).$$

Let $d_{\max} = \max\{\pi_1(P(q)) \mid q \in Q\}$ and $d_{\min} = \min\{\pi_1(P(q)) \mid q \in Q\}$. Then, for every $k, 0 \leq k < m + \ell n$,

$$\mathsf{Prob}[Y_k \neq o_k] \leq r e^{-d_{\min}\epsilon}(1 + \frac{d_{\max}\epsilon}{2})$$

Using the union rule of probabilities, we see that,

$$\mathsf{Prob}[\exists k < m + \ell n, Y_k \neq o_k] \leq r(m + \ell n) e^{-d_{\min}\epsilon}(1 + \frac{d_{\max}\epsilon}{2}).$$

Given $\ell > 0$, let $\epsilon_\ell \in \mathbb{R}$, be the smallest value such that

$$\forall \epsilon \geq \epsilon_\ell,\ r(m + \ell n) e^{-d_{\min}\epsilon}(1 + \frac{d_{\max}\epsilon}{2}) \leq \frac{1}{2}.$$

Now,

$$\mathsf{Pr}[\epsilon, \rho_\alpha(\ell)] = 1 - \mathsf{Prob}[\exists k < m + \ell n,\ Y_k \neq o_k].$$

From the construction of ϵ_ℓ and above observations, we see that $\forall \epsilon \geq \epsilon_\ell,\ \mathsf{Pr}[\epsilon, \rho_\alpha(\ell)] \geq \frac{1}{2}$.

Now, recall the integers i, j fixed earlier. Intuitively, we define $\beta(\ell)$ so that each of the guards in the transitions $t_{j+\ell' n}, 0 \leq \ell' < \ell$ are satisfied with probability $< \frac{1}{2}$. For each ℓ', $0 \leq \ell' < \ell$, we let $b_{i+\ell' n} = a_{j+\ell' n} + \mu_j - \mu_i$ and $b_{j+\ell' n} = a_{i+\ell' n} + \mu_i - \mu_j$. We observe the following. Now, for each ℓ', $0 \leq \ell' < \ell$, the following hold. $c_{j+\ell' n} = c_j \neq \mathsf{true}$. If $c_{j+\ell' n}$ is the guard $\mathsf{insample} \geq \mathsf{x}$ then $b_{i+\ell' n} + \mu_i = b_{j+\ell' n} + \mu_j + 1$ since $a_{j+\ell' n} + \mu_j = a_{i+\ell' n} + \mu_i + 1$. If $c_{j+\ell' n}$ is the guard $\mathsf{insample} < \mathsf{x}$ then $b_{j+\ell' n} + \mu_j = b_{i+\ell' n} + \mu_i + 1$ since $a_{i+\ell' n} + \mu_i = a_{j+\ell' n} + \mu_j + 1$. We define $b_{i'}$, for all values of $i' < m + \ell n$ and $i' \notin \{i + \ell' n, j + \ell' n \mid 0 \leq \ell' < \ell\}$, so that $\beta(\ell)$ is a neighbour of $\alpha(\ell)$. It is not difficult to see that such a sequence $\beta(\ell)$ can be defined. Let $\rho_\beta(\ell)$ be the path such that $\mathsf{abstract}(\rho_{\beta(\ell)}) = \eta(\ell)$ and $\mathsf{inseq}(\rho_\beta(\ell)) = \beta(\ell)$.

For each $k, 0 \leq k < m + \ell n$, let U_k be the random variable with distribution given by $\mathsf{Lap}(d_{q_k}\epsilon, b_k)$ and Z_k be denoting the k^{th} output of η on the input sequence $\beta(\ell)$. Let $d' = \min(d_i, d_j)$ and $d'' = \max(d_i, d_j)$. Now, $\mathsf{Prob}[Z_j = o_j]$ is given by $\mathsf{Prob}[U_j \geq U_i]$ if c_j is the guard $\mathsf{insample} \geq \mathsf{x}$, otherwise it is given by $\mathsf{Prob}[U_j \leq U_i]$. Using Lemma 5 and similar reasoning as given earlier, we see that

$$\mathsf{Prob}[Z_j = o_j] \leq r' e^{-d'\epsilon}(1 + \frac{d''\epsilon}{2})$$

for some constant r'. For each ℓ', $0 < \ell' < \ell$, using the same reasoning as above with the random variables $U_{i+\ell' n}, U_{j+\ell' n}$, we see that

$$\mathsf{Prob}[Z_{j+\ell' n} = o_{j+\ell' n}] \leq r' e^{-d'\epsilon}(1 + \frac{d''\epsilon}{2}).$$

Since for any ℓ_1, ℓ_2 such that $0 \leq \ell_1 < \ell_2 < \ell$, the random variables $U_{i+\ell_1 n}, U_{j+\ell_1 n}$ are independent of $U_{i+\ell_2 n}, U_{j+\ell_2 n}$, we see that

$$\mathsf{Prob}[\forall \ell', 0 \leq \ell' < \ell, Z_{j+\ell' n} = o_{j+\ell' n}] \leq {r'}^\ell e^{-d'\ell\epsilon}(1 + \frac{d''\epsilon}{2})^\ell.$$

Thus,

$$\mathsf{Prob}[\forall k, 0 \leq k < m + \ell n, Z_k = o_k] \leq {r'}^\ell e^{-d'\ell\epsilon}(1 + \frac{d''\epsilon}{2})^\ell.$$

The LHS of the above equation is exactly $\mathsf{Pr}[\epsilon, \rho_\beta(\ell)]$.

Thus, for any $\ell > 0$,

$$\forall \epsilon \geq \epsilon_\ell,\ \frac{\mathsf{Pr}[\epsilon, \rho_\alpha(\ell)]}{\mathsf{Pr}[\epsilon, \rho_\beta(\ell)]} \geq \frac{1}{2}(\frac{e^{d'\epsilon}}{r'(1 + \frac{d''\epsilon}{2})})^\ell.$$

We claim that for any $s > 0$, $\exists \ell, \epsilon$ such that

$$\frac{1}{2}(\frac{e^{d'\epsilon}}{r'(1 + \frac{d''\epsilon}{2})})^\ell > e^{s\epsilon}.$$

Now the above inequality holds if

$$\frac{e^{(d'\ell - s)\epsilon}}{(1 + \frac{d'\epsilon}{2})^\ell} > 2{r'}^\ell.$$

Choose ℓ so that $d'\ell > s$. Since the denominator of the left hand side term of the last inequality grows polynomially in ϵ, while its numerator grows exponentially in ϵ, it is easy to see that $\exists \epsilon_0 > \epsilon_\ell$ such that

$$\forall \epsilon \geq \epsilon_0,\ \frac{e^{(d'\ell - s)\epsilon}}{(1 + \frac{d'\epsilon}{2})^\ell} > 2{r'}^\ell.$$

The crucial observation we now make is that, thanks to output determinism, for every input sequence α and output sequence γ, there is at most one path $\rho_{\alpha,\gamma}$ such that $\mathsf{inseq}(\rho_{\alpha,\gamma}) = \alpha$ and $\mathsf{outseq}(\rho_{\alpha,\gamma}) = \alpha$. This observation combined with the above inequality shows that $\mathcal{A}$ is not differentially private.

Leaking pairs implies no privacy

Lemma 7. *A DiPA $\mathcal{A}$ is not differentially private if it has a leaking pair of cycles (C, C') such that C is reachable from the initial state of $\mathcal{A}$.*

Proof. Thanks to Lemma 6, we can assume $\mathcal{A}$ does not have a leaking cycle. Let $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$. Assume that $\mathcal{A}$ has a leaking pair of cycles (C, C') such that C is reachable from q_{init}. Assume that C is an L-cycle and C' is a G-cycle. (The proof for the case when C is a G-cycle and C' is an L-cycle is similar but symmetric and is left out). Thanks to our assumption that we do not have leaking cycles, it means that both C, C' do not have assignment transitions. We further assume that C, C' are distinct. If they are the same then it is straightforward to prove that $\mathcal{A}$ is not differentially private, using more or less the same proof. We also assume that all the states in $\mathcal{A}$ are input states. The case when $\mathcal{A}$ has both input and non-input states can also be proved using more or less the same proof.

Let the lengths of C, C' be n_1, n_2, respectively. Now, for any $\ell > 0$, consider the following abstract path η_ℓ in $\mathcal{A}$ starting from q_{init} in which the cycles C, C' are repeated ℓ times each. The path

$$\begin{aligned}\eta_\ell \quad = \quad & q_0\sigma_0 \cdots q_u\sigma_u \cdots q_v\sigma_v \cdots q_{v+n_1\ell-1}\sigma_{v+n_1\ell-1} \cdots \\ & \cdots q_w\sigma_w \cdots o_{w+n_2\ell-1}q_{w+n_2\ell}\end{aligned}$$

where the following guards are satisfied. For each k, let t_k be the k-th transition of η. and c_k be the gaurd of the k-th transition.

1) $q_0 = q_{\mathsf{init}}$
2) $q_v\sigma_v q_{v+1}\sigma_{v+1} \cdots q_{v+n_1-1}\sigma_{v+n_1-1}q_{v+n_1}$ is the cycle C
3) $t_{j+n_1} = t_j$ for all j, $v \leq j < v + n_1(\ell - 1)$
4) $q_w\sigma_w q_{w+1}\sigma_{w+1} \cdots q_{w+n_2-1}\sigma_{w+n_2-1}q_{w+n_2}$ is the cycle C'
5) $t_{j+n_2} = t_j$ for all j, $w \leq j < w + n_2(\ell - 1)$
6) t_u is an assignment transition and $\forall\, j, u < j < v + n_1\ell$ and $\forall\, j, j \geq w$, t_j is a non-assignment transition
7) for all j, $v + n_1\ell \leq j < w$, if t_j is an assignment transition then c_j is the guard $\mathsf{insample} \geq \mathsf{x}$.

Observe that the last assignment transition before $t_{v+n_1\ell}$ is t_u, all assignment transitions from $t_{v+n_1\ell}$ up to t_w have $\mathsf{insample} \geq \mathsf{x}$ as their guard, the segment of the path from t_v to $t_v + n_1\ell - 1$ is the part where cycle C is repeated ℓ times and the segment of the path from t_w to $t_w + n_2\ell - 1$ is the part where cycle C' is repeated ℓ times. Let d_k and μ_k be such that $P(q_k) = (d_k, \mu_k)$ for each k. We have that $c_0 = \mathsf{true}$ and t_0 is an assignment transition.

Let $\gamma(\ell) = o_0 \cdots o_{m+\ell n-1}$ be the output sequence of length $m + \ell n$ such that $o_k = \sigma_k$ if $\sigma_k \in \Gamma$, otherwise $o_k = (\sigma_k, -\infty, \infty)$. Once again, we let t_k be the k-th transition of η_ℓ and c_k be the guard of the k-th transition. Now, given $\ell > 0$, we define two neighboring input sequences $\alpha(\ell) = a_0 \cdots a_{m+\ell n-1}$ and $\beta(\ell) = b_0 \cdots b_{m+\ell n-1}$ each of length $m + \ell n$.

Now, we define two adjacent input sequences $\alpha(\ell) = a_0 \cdots a_{w+n_2\ell-1}$ and $\beta(\ell) = b_0 \cdots b_{w+n_2\ell-1}$ as follows. For all $j, 0 \leq j < v$ and for all $j, v+n_1\ell \leq j < w$,$a_j = b_j = 0$; for all j, $v \leq j < v + n_1\ell$ and for all j, $w \leq j < w + n_2\ell$, if c_j is the guard $\mathsf{insample} \geq \mathsf{x}$ then $a_j = \frac{1}{2} - \mu_j$, $b_j = -\frac{1}{2} - \mu_j$, if c_j is the guard $\mathsf{insample} < \mathsf{x}$ then $a_j = -\frac{1}{2} - \mu_j$, $b_j = \frac{1}{2} - \mu_j$ and if c_j is true then $a_j = b_j = 0$. It is not difficult to see that $\alpha(\ell)$ and $\beta(\ell)$ are adjacent. Let $\rho_\alpha(\ell)$ be the path such that $\mathsf{abstract}(\rho_{\alpha(\ell)}) = \eta(\ell)$ and $\mathsf{inseq}(\rho_\alpha(\ell)) = \alpha(\ell)$. Let $\rho_\beta(\ell)$ be the path such that $\mathsf{abstract}(\rho_{\beta(\ell)}) = \eta(\ell)$ and $\mathsf{inseq}(\rho_\beta(\ell)) = \beta(\ell)$.

Let X_j, U_j be random variables with distributions given by $\mathsf{Lap}(d_j\epsilon, a_j + \mu_j)$ and $\mathsf{Lap}(d_j\epsilon, b_j + \mu_j)$, respectively. Observe that t_u is the last assignment transition in η_ℓ. For each $j > u$, for any given $y \in \mathbb{R}$, let $g_j(y), h_j(y)$ be the probabilities defined as follows: if c_j is the guard $\mathsf{insample} \geq \mathsf{x}$ then $g_j(y) = \mathsf{Prob}[X_j \geq y]$ and $h_j(y) = \mathsf{Prob}[U_j \geq y]$; if c_j is the guard $\mathsf{insample} < \mathsf{x}$ then $g_j(y) = \mathsf{Prob}[X_j < y]$ and $h_j(y) = \mathsf{Prob}[U_j < y]$; if c_j is true then $g_j(y) = h_j(y) = 1$. It should be easy to see that, for all j, $u < j < v$ and for all j, $v + n_1\ell \leq j < w$, $a_j = b_j$ and hence $g_j(y) = h_j(y)$. Now, we have the following claim.

Claim: For all $j, v \leq j < v + n_1\ell$, and for all j, $w \leq j < w + n_2\ell$, it is the case that $g_j(y) \geq h_j(y)$ for all $y \in \mathbb{R}$, and the following additional inequalities hold.

1) If $y \leq 0$ and c_j is the guard $\mathsf{insample} < \mathsf{x}$ then $g_j(y) \geq e^{\frac{1}{2}d_j\epsilon}h_j(y)$.
2) If $y > 0$ and c_j is the guard $\mathsf{insample} \geq \mathsf{x}$ then $g_j(y) \geq e^{\frac{1}{2}d_j\epsilon}h_j(y)$.

Proof. Observe that when $c_j = \mathsf{true}$ then trivially $g_j(y) = h_j(y)$. Now, consider the case when $y < -\frac{1}{2}$. If c_j is the guard $\mathsf{insample} \geq \mathsf{x}$ then $g_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(\frac{1}{2}-y)}$ and $h_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(-\frac{1}{2}-y)}$(this is so since $a_j + \mu_j = \frac{1}{2}$ and $b_j + \mu_j = -\frac{1}{2}$) ; in this case $\frac{1}{2} - y > -\frac{1}{2} - y$ and hence $g_j(y) \geq h_j(y)$. If c_j is the guard $\mathsf{insample} < \mathsf{x}$ then $g_j(y) = \frac{1}{2}e^{-d_j\epsilon(-\frac{1}{2}-y)}$ and $h_j(y) = \frac{1}{2}e^{-d_j\epsilon(\frac{1}{2}-y)}$; from this we see that $g_j(y) \geq e^{d_j\epsilon}h_j(y)$.

Now consider the case when $y \in [-\frac{1}{2}, 0]$. If c_j is the guard $\mathsf{insample} \geq \mathsf{x}$ then $g_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(\frac{1}{2}-y)}$ and $h_j(y) = \frac{1}{2}e^{-d_j\epsilon(y+\frac{1}{2})}$; since $g_j(y) \geq \frac{1}{2}$ and $h_j(y) \leq \frac{1}{2}$, we see that $g_j(y) \geq h_j(y)$. If c_j is the guard $\mathsf{insample} < x$ then $g_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(y+\frac{1}{2})}$ and $h_j(y) = \frac{1}{2}e^{-d_j\epsilon(\frac{1}{2}-y)}$; since $g_j(y) \geq \frac{1}{2}$, we see that $g_j(y) \geq e^{\frac{1}{2}d_j\epsilon}h_j(y)$.

Now consider the case when $y > 0$. If $y \leq \frac{1}{2}$ and c_j is $\mathsf{insample} \geq \mathsf{x}$ then $g_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(\frac{1}{2}-y)}$ and $h_j(y) = \frac{1}{2}e^{-d_j\epsilon(y+\frac{1}{2})}$; observe that $g_j(y) \geq \frac{1}{2}$ and $h_j(y) \leq \frac{1}{2}e^{-\frac{1}{2}d_j\epsilon}$; from this we get the desired inequality.

If $y \leq \frac{1}{2}$ and c_j is $\mathsf{insample} < \mathsf{x}$ then $g_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(y+\frac{1}{2})}$ and $h_j(y) = \frac{1}{2}e^{-d_j\epsilon(\frac{1}{2}-y)}$; since $g_j(y) \geq \frac{1}{2}$ and $h_j(y) \leq \frac{1}{2}$, we see $g_j(y) \geq h_j(y)$. If $y > \frac{1}{2}$ and c_j is $\mathsf{insample} \geq \mathsf{x}$ then $g_j(y) = \frac{1}{2}e^{-d_j\epsilon(y-\frac{1}{2})}$ and $h_j(y) = \frac{1}{2}e^{-d_j\epsilon(y+\frac{1}{2})}$; from this we see that the desired inequality follows easily. If $y > \frac{1}{2}$ and c_j is $\mathsf{insample} < \mathsf{x}$ then $g_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(y+\frac{1}{2})}$ and $h_j(y) = 1 - \frac{1}{2}e^{-d_j\epsilon(y-\frac{1}{2})}$; it is easy to see that $g_j(y) \geq h_j(y)$. □

Let $S_1(\ell)$ be the set of all j such that $v \leq j < v + n_1\ell$ and c_j is the guard $\mathsf{insample} < x$. Let $S_2(\ell)$ be the set of all j such that $w \leq j < w + n_2\ell$, and c_j is the guard $\mathsf{insample} \geq \mathsf{x}$. Since C is an L-cycle and C' is a G-cycle, we see that the cardinalities of both $S_1(\ell)$ and $S_2(\ell)$ are $\geq \ell$. Let $d_{\min} = \min\{d_j \mid j \in S_1(\ell) \cup S_2(\ell)\}$. Clearly $d_{\min} > 0$.

Let $\rho_\alpha(\ell)$ be the path such that $\mathsf{abstract}(\rho_{\alpha(\ell)}) = \eta(\ell)$ and $\mathsf{inseq}(\rho_\alpha(\ell)) = \alpha(\ell)$. For $k \leq q + n_2\ell$, let $\rho_\alpha(\ell)||k$ (resp, $\rho_\beta(\ell)||k$) be the suffix of $\rho_\alpha(\ell)$ (resp. $\rho_\alpha(\ell)||k$) starting with q_k.

Since C' is a G-cycle, from the above claim, we see that $\forall y \in \mathbb{R}$, $\mathsf{Pr}[\rho_\alpha(\ell)||w, y] \geq \mathsf{Pr}[\rho_\beta(\ell)||w, y]$, and $\forall y > 0$, $\mathsf{Pr}[\rho_\alpha(\ell)||w, y] \geq e^{\frac{1}{2}d_{\min}\ell\epsilon}\mathsf{Pr}[\rho_\beta(\ell)||w, y]$. Using the above property and the previous claim, together with the assumption that $\forall j,\ v + n_1\ell \leq j < w$, if t_j is an assignment transition then it's guard is $\mathsf{insample} \geq \mathsf{x}$, the following can be proved by downward induction on k, $\forall k,\ v + n_1\ell \leq k < w$: $\forall y \in \mathbb{R}$, $\mathsf{Pr}[\rho_\alpha(\ell)||k, y] \geq \mathsf{Pr}[\rho_\beta(\ell)||k, y]$, and $\forall y > 0$, $\mathsf{Pr}[\rho_\alpha(\ell)||k, y] \geq e^{\frac{1}{2}d_{\min}\ell\epsilon}\mathsf{Pr}[\rho_\beta(\ell)||k, y]$.

Now, it should be easy to see that $\forall y \in \mathbb{R}$,

$$\mathsf{Pr}[y, \rho_\alpha(\ell)||v] = (\textstyle\prod_{v \leq j < v+n_1\ell} g_j(y))\mathsf{Pr}[y, \rho_\alpha(\ell)||v + n_1\ell]$$
$$\mathsf{Pr}[y, \rho_\beta(\ell)||v] = (\textstyle\prod_{v \leq j < v+n_1\ell} h_j(y))\mathsf{Pr}[y, \rho_\beta(\ell)||v + n_1\ell].$$

Observe that $\forall j,\ v \leq j < v + n_1\ell$,

$$\begin{aligned}
&\forall y \leq 0: \quad g_j(y) \geq e^{\frac{1}{2}d_{\min}\epsilon}h_j(y),\\
&\qquad\qquad \mathsf{Pr}[y, \rho_\alpha(\ell)||j] \geq \mathsf{Pr}[y, \rho_\beta(\ell)||j]\\
&\text{and}\\
&\forall y > 0: \quad g_j(y) \geq h_j(y),\\
&\qquad\qquad \mathsf{Pr}[y, \rho_\alpha(\ell)||j] \geq e^{\frac{1}{2}d_{\min}\ell\epsilon}\mathsf{Pr}[y, \rho_\beta(\ell)||j].
\end{aligned}$$

From this we get the following:

$$\forall y \in \mathbb{R},\ \mathsf{Pr}[y, \rho_\alpha(\ell)||v] \geq e^{\frac{1}{2}d_{\min}\ell\epsilon}\mathsf{Pr}[y, \rho_\beta(\ell)||v].$$

Using this we can show by the definition of probability of a path that

$$\frac{\mathsf{Pr}[\epsilon, \rho_\alpha(\ell)]}{\mathsf{Pr}[\epsilon, \rho_\beta(\ell)]} \geq e^{\frac{1}{2}d_{\min}\ell\epsilon}.$$

Since ℓ can be made arbitrarily large, we see that $\mathcal{A}$ is not $d\epsilon$-differentially private, for any $d > 0$. Hence $\mathcal{A}$ is not differentially private. □

Disclosing cycles implies no privacy

Lemma 8. *A DiPA $\mathcal{A}$ is not differentially private if it has a reachable disclosing cycle.*

Proof. Thanks to Lemma 6 and Lemma 7, we can assume $\mathcal{A}$ does not have leaking cycles or leaking pairs. Assume that $\mathcal{A}$ is well-formed, but there is a reachable disclosing cycle C in $\mathcal{A}$ that has a transition whose output is $\mathsf{insample}$. The proof for the case when C has a transition whose output is $\mathsf{insample}'$ is simpler and is left out. Now, if the transition of C whose output is $\mathsf{insample}$ has the guard true, then it can be shown easily that repeating the cycle ℓ times incurs a privacy cost linear in $\ell\epsilon$, and hence $\mathcal{A}$ cannot be $d\epsilon$-differentially private for any $d > 0$. Thus, we consider more interesting case when the guard is $\mathsf{insample} < \mathsf{x}$ or $\mathsf{insample} \geq \mathsf{x}$.

We consider the case when C has a transition with output $\mathsf{insample}$. Since $\mathcal{A}$ is well-formed the cycle C has no assignment transitions. Let $\eta = q_0\sigma_0q_1\sigma_1 \cdots q_{j+m-1}\sigma_{j+m-1}q_{j+m}$ for $k = 0, \ldots, j+m-1$ be an abstract path such that $q_0 = q_{\mathsf{init}}$, $q_j = q_{j+m}$, and the final m transitions of ρ is the abstract cycle corresponding to C. Fix $0 \leq r < m$ be such that $\sigma_{j+r} = \mathsf{insample}$. We assume that the guard of the $(j+r)$-th transition is $\mathsf{insample} \geq \mathsf{x}$. The case when it is $\mathsf{insample} < \mathsf{x}$ is similar and left out. Further, let d_k and μ_k be such that $P(q_k) = (d_k, \mu_k)$ for each k.

Fix $\ell > 0$. We define an abstract path η_ℓ starting from q_{init} by repeating the cycle C ℓ times. Formally, $\eta_\ell = q_0\sigma_0q_1\sigma_1 \cdots q_{j+\ell m-1}\sigma_{j+\ell m-1}q_{j+\ell m}$ such that $q_k = q_{k-m}$ and $\sigma_k = \sigma_{k-n}$ for $j + m \leq k \leq j + \ell m$. Let t_k be the k-th transition of η_ℓ and c_k be the guard of the k-th transition. We have that $\sigma_{j+nm+r} = \mathsf{insample}$, for all n such that $0 \leq n < \ell$.

Now we construct two input sequences $\alpha(\ell) = a_0 \cdots a_{j+\ell m-1}$ and $\beta(\ell) = b_0 \cdots b_{j+\ell m-1}$ as follows. We take $a_k = -\mu_k$, for all $k, 0 \leq k < j + \ell m$ such that t_k is an input transition, otherwise we take $a_k = \tau$. We take $b_k = -\mu_k - 1$ if $k = j+nm+r$ for some $0 \leq n < \ell$ and $b_k = a_k$ otherwise. Let $\rho(\ell) = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{j+\ell m-1} \xrightarrow{a_{j+\ell m-1},o_{j+\ell m-1}} q_{j+\ell m}$ be the path such that

- $\eta = \mathsf{abstract}(\rho(\ell))$,
- $\mathsf{inseq}(\rho(\ell)) = \alpha(\ell)$, and
- all k, i) $o_k = \sigma_k$ if $\sigma_k \in \Gamma$, ii) $o_k = (\sigma_k, 0, \infty)$ if $k = j + nm + r$ for some $0 \leq n < \ell$, and iii) $o_k = (\sigma_k, -\infty, \infty)$ otherwise.

Let $\rho'(\ell) = q_0 \xrightarrow{b_0,o_0} q_1 \xrightarrow{b_1,o_1} q_2 \cdots q_{j+\ell m-1} \xrightarrow{b_{j+\ell m-1},o_{j+\ell m-1}} q_{j+\ell m}$ be the path that is equivalent to ρ and $\mathsf{inseq}(\rho'(\ell)) = \beta(\ell)$.

Let $\rho(\ell)||k$ and $\rho'(\ell)||k$ be the suffixes of executions $\rho(\ell)$ and $\rho'(\ell)$ starting from state q_k. Using backward induction, we can easily show that for each x_0, $\mathsf{Pr}[x_0, \rho(\ell)||k], \mathsf{Pr}[x_0, \rho'(\ell)||k]$ are non-zero and that

$$\mathsf{Pr}[x_0, \rho(\ell)||k] = e^{\#(k)d_{j+r}\epsilon}\mathsf{Pr}[x_0, \rho'(\ell)||k]$$

where $\#(k)$ is the number of indices k_1 such that $k \leq k_1 < j + m\ell - 1$ and $k_1 = j + nm + r$ for some $0 \leq n < \ell$. Thus,

$$\mathsf{Pr}[\epsilon, \rho(\ell)] = e^{\ell d_{j+r}\epsilon}\mathsf{Pr}[\epsilon, \rho'(\ell)].$$

Now, ℓ is arbitrary and hence for every $d > 0$, there is an ℓ such that $\mathsf{Pr}[\epsilon, \rho(\ell)] > e^{d\epsilon}\mathsf{Pr}[\epsilon, \rho'(\ell)]$. Hence $\mathcal{A}$ is not differentially private. □

Privacy violating paths implies no privacy

Lemma 9. *A DiPA $\mathcal{A}$ is not differentially private if it has a reachable privacy violating path.*

Proof. Thanks to Lemma 6, Lemma 8 and Lemma 7, we can assume $\mathcal{A}$ does not have leaking cycles, disclosing cycles or leaking pairs. We give the proof for one of the cases of a privacy violating path, where the path starts with a transition whose guard is $\mathsf{insample} < \mathsf{x}$ and which lies on an L-cycle C which is followed by an AG-path ending in a transition

with guard $\mathsf{insample} \geq \mathsf{x}$ and whose output is $\mathsf{insample}$. (The proofs for other cases of the privacy violating path are similar and are leftout.) Since $\mathcal{A}$ is well-formed, the cycle C does not have an assignment transition.

Fix $\ell > 0$. Consider an abstract path $\eta(\ell) = q_0\sigma_0 q_1\sigma_1 \cdots q_{n-1}\sigma_{n-1}q_n$ of length n from the initial state q_{init} such that $\eta(\ell)$ contains the cycle C repeated ℓ times, and upon exiting the cycle continues onto the AG-path p such that the last transition of the AG-path has guard $\mathsf{insample} \geq \mathsf{x}$ and outputs $\mathsf{insample}$. Fix a transition of C with guard $\mathsf{insample} < \mathsf{x}$, and let $k_1, k_2, \ldots, k_\ell$ be the indices where this transition occurs in $\eta(\ell)$. Let $P(q_k) = (d_k, \mu_k)$. Next, we construct two input sequences $\alpha(\ell) = a_0 \cdots a_n$ and $\beta(\ell) = b_0 \cdots b_n$ of length n as follows. If the kth transition of $\eta(\ell)$ is a non-input transition then $a_k = b_k = \tau$. If $k \in \{k_1, k_2, \ldots, k_\ell\}$ then $a_k = -\mu_k$ and $b_k = -\mu_k + 1$. For all other ks, $a_k = b_k = -\mu_k$. Let $\rho(\ell) = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{j+\ell m-1} \xrightarrow{a_{j+\ell m-1},o_{j+\ell m-1}} q_{j+\ell m}$ be the path such that

- $\eta = \mathsf{abstract}(\rho(\ell))$,
- $\mathsf{inseq}(\rho(\ell)) = \alpha(\ell)$, and
- for all k, i) $o_k = \sigma_k$ if $\sigma_k \in \Gamma$, ii) $o_k = (\sigma_k, -\infty, 0)$ if $k = n$, and iii) $o_k = (\sigma_k, -\infty, \infty)$ otherwise.

Let $\rho'(\ell) = q_0 \xrightarrow{b_0,o_0} q_1 \xrightarrow{b_1,o_1} q_2 \cdots q_{j+\ell m-1} \xrightarrow{b_{j+\ell m-1},o_{j+\ell m-1}} q_{j+\ell m}$ ibe the path that is equivalent to ρ and $\mathsf{inseq}(\rho'(\ell)) = \beta(\ell)$.

Please note that in $\rho(\ell), \rho'(\ell)$, the last output is a non-positive number. As the path p is also an AG-path, this implies that stored value of x during the ℓ executions of C is also a non-positive number. Combined with the fact that C is an L-cycle and the construction of $\rho(\ell), \rho'(\ell)$, it can be shown that

$$\mathsf{Pr}[\epsilon, \rho(\ell)] = e^{\ell d_{k_1}\epsilon}\mathsf{Pr}[\epsilon, \rho'(\ell)].$$

As in the case of disclosing cycle (See Lemma 8), we can conclude that $\mathcal{A}$ is not differentially private. □

Appendix C
Sufficiency of well-formedness

We shall now show that if the DiPA $\mathcal{A}$ is well-formed then $\mathcal{A}$ is differentially private, thus establishing the "if" part of Theorem 2. Please note that it suffices to prove Lemma 3. In order to manage complexity, we shall first prove the Lemma for the case that $\mathcal{A}$ outputs only elements of the discrete set Γ (See Lemma 11). Then we shall tackle the case of all outputs (See Lemma 13). Please note that we shall use the notions of path suffixes and abstract paths introduced in Appendix A.

Before we proceed, we need a technical lemma.

Lemma 10. *Let f and g_i for $i = 1, \ldots, k$ be non-negative functions from $\mathbb{R}$ to $\mathbb{R}$, i.e., $f(y), g_i(y) \geq 0$ for all i, y. For $i = 1, \ldots, k$, let $\theta_i \in [-1, 1]$. Let $x_0, x_1 \in \mathbb{R} \cup \{\infty, -\infty\}$, be such that $x_0 < x_1$. Then, the following inequalities are satisfied for all $k \geq 0$. The empty products (the case when $k = 0$) in these inequalities are taken be 1.*

1. $\int_{x_0}^{x_1} f(x) \prod_{i=1}^{k} \int_{x}^{\infty} g_i(y - \theta_i) dy dx \geq \int_{x_0+1}^{x_1+1} f(x-1) \prod_{i=1}^{k} \int_{x}^{\infty} g_i(y) dy\, dx$

2. $\int_{x_0}^{x_1} f(x) \prod_{i=1}^{k} \int_{-\infty}^{x} g_i(y - \theta_i) dy dx \geq \int_{x_0-1}^{x_1-1} f(x+1) \prod_{i=1}^{k} \int_{-\infty}^{x} g_i(y) dy\, dx$

Proof. We prove the inequality (1) as follows. For each $i = 1, \ldots, k$, by substituting $z = y - \theta_i$, we get $\int_x^\infty g_i(y-\theta_i)dy = \int_{x-\theta_i}^\infty g_i(z)dz$. Since $\theta_i \in [-1, 1]$ and g_i is a positive function, we get $\int_{x-\theta_i}^\infty g_i(z)dz \geq \int_{x+1}^\infty g_i(z)dz$. By rewriting the left hand side of the inequality (1) as specified above and by substituting, $u = x + 1$, we get the right hand side of the inequality (1) where the outer integral is over the variable u. By replacing u by x and z by y, we get the right hand side of the inequality.

We prove the inequality (2) as follows. As before, for each $i = 1, \ldots, k$, we rewrite the integral $\int_{-\infty}^{x} g_i(y - \theta_i)dy$ as $\int_{-\infty}^{x-\theta_i} g_i(z)dz$ and then observe that this is $\geq \int_{-\infty}^{x-1} g_i(z)dz$. Substituting $u = x - 1$, and then replacing u by x later, we get the inequality (2). □

DiP automata with Finite Outputs

Lemma 11. *Let $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$ be a well-formed DiPA with finite outputs. Let ρ be a path of length $n > 0$ such that the initial transition (i.e. the 0th transition), t_0, of ρ is an assignment transition. Let c_0 be the guard of t_0. Let ρ' be a path that is equivalent to ρ such that $\mathsf{inseq}(\rho')$ is a neighbor of $\mathsf{inseq}(\rho)$. Then the following properties hold for all $x_0 \in \mathbb{R}$.*

1) *If the guard c_0 is $\mathsf{insample} \geq \mathsf{x}$, and the first cycle transition in ρ is a G-cycle transition and no assignment transition with guard $\mathsf{insample} < \mathsf{x}$ appears before it, then*
$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 + 1, \rho].$$

2) *If the guard c_0 is $\mathsf{insample} \geq \mathsf{x}$ and one of the following holds: (a) ρ has no cycle transitions, (b) the first cycle transition in ρ is a G-cycle transition and an assignment transition with guard $\mathsf{insample} < \mathsf{x}$ appears before it, (c) the first cycle transition in ρ is an L-cycle transition, then*
$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 - 1, \rho].$$

3) *If the guard c_0 is $\mathsf{insample} < \mathsf{x}$ and the first cycle transition in ρ is a L-cycle transition and no assignment transition with guard $\mathsf{insample} \geq \mathsf{x}$ appears before it, then*
$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 - 1, \rho].$$

4) *If the guard c_0 is $\mathsf{insample} < \mathsf{x}$ and one of the following holds: (a) ρ has no cycle transitions, (b) the first cycle transition in ρ is a L-cycle transition and an assignment transition with guard $\mathsf{insample} \geq \mathsf{x}$ appears before it, (c) the first cycle transition in ρ is a G-cycle transition, then*
$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 + 1, \rho].$$

5) *If the guard* c_0 *is* true*, then* $\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0, \rho]$.

Proof. Let $\rho = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$ and $\rho' = q_0 \xrightarrow{b_0,o_0} q_1 \xrightarrow{b_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{b_{n-1},o_{n-1}} q_n$. Let $t_0, \ldots, t_{n-1}$ be the transitions of ρ and let $c_0, \ldots, c_{n-1}$ be their respective guards. For each $k \leq n$, let d_k, μ_k be such that $P(q_k) = (d_k, \mu_k)$. Recall that, for any k, $\rho||k$ denotes the suffix of ρ starting from q_k. We assume that there are no cycle transitions that are assignments. This is because if there is a cycle with an assignment then the guards on all other transitions must be true. Hence, we can never exit the cycle. Further, it is easy to see that this cycle has the same "behavior" in both ρ and ρ'.

For each k, such that $0 \leq k < n$, let g_k, g'_k, θ_k be functions of a single variable given by

$$g_k(y) = \begin{cases} \frac{d_k\epsilon}{2}e^{-d_k\epsilon|y-a_k-\mu_k|} & t_i \text{ is an input transition} \\ \frac{d_k\epsilon}{2}e^{-d_k\epsilon|y-\mu_k|} & \text{otherwise} \end{cases},$$

$$g'_k(y) = \begin{cases} \frac{d_k\epsilon}{2}e^{-d_k\epsilon|y-b_k-\mu_k|} & t_i \text{ is an input transition} \\ \frac{d_k\epsilon}{2}e^{-d_k\epsilon|y-\mu_k|} & \text{otherwise} \end{cases}$$

and

$$\theta_k = \begin{cases} b_k - a_k & t_i \text{ is an input transition} \\ 0 & \text{otherwise.} \end{cases}$$

Observe that, for each $k \geq 0$, $g'_k(y) = g_k(y - \theta_k)$. Since $|\theta_k| \leq 1$, we see that $g'_k(y) \geq e^{-d_k\epsilon}g_k(y)$, for all $y \in \mathbb{R}$.

We prove the lemma by induction on the number of assignment transitions in ρ.

Base Case: In the base case, ρ has one assignment transition which is t_0. Let S_1 and S_2 be the sets of $k > 0$ such that c_k is $\mathsf{insample} \geq \mathsf{x}$ and c_k is $\mathsf{insample} < \mathsf{x}$, respectively. Now, assume the condition of statement (1) of the Lemma is satisfied. Observe that S_1 includes all G-cycle transitions whose guard is $\mathsf{insample} \geq \mathsf{x}$. Observe that, since $\mathcal{A}$ is well-formed, for all $k \in S_2$, t_k does not lie on a cycle and hence is a critical transition. Similarly t_0 is also a critical transition. Now, we see that

$$\mathsf{Pr}[x_0, \rho'] = \int_{x_0}^{\infty} f(x) \prod_{k \in S_1} \int_x^{\infty} g'_k(y)dy\, dx$$

where $f(x) = g'_0(x) \prod_{k \in S_2} \int_{-\infty}^{x} g'_k(y)dy$. Now, substituting $g'_k(y) = g_k(y - \theta_k)$ (for $k \in S_1$) in the above equation and using inequality (1) of Lemma 10, we see that

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0+1}^{\infty} f(x-1) \prod_{k \in S_1} \int_x^{\infty} g_k(y)dy\, dx.$$

Observe that

$$f(x-1) = g_0(x - (1 + \theta_0)) \prod_{k \in S_2} \int_{-\infty}^{x-1} g_k(y - \theta_k)dy.$$

Now, by introducing a new variable z such that $z = y + 1$, we see that

$$\int_{-\infty}^{x-1} g_k(y - \theta_k)dy = \int_{-\infty}^{x} g_k(z - (1 + \theta_k))dz.$$

From this, it is easy to see that

$$f(x-1) \geq e^{-2(d_0+\sum_{k\in S_2} d_k)\epsilon} g_0(x) \prod_{k \in S_2} \int_{-\infty}^{x} g_k(y)dy.$$

Observe that $\mathsf{wt}(\rho) \geq 2(d_0 + \sum_{k \in S_2} d_k)$. Putting all the above observations together, we get

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon} \int_{x_0+1}^{\infty} g_0(x) \prod_{k \in S_2} \int_{-\infty}^{x} g_k(y)dy \prod_{k \in S_1} \int_x^{\infty} g_k(y)dy.$$

Observe that the right hand side of the above inequality is $e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 + 1, \rho]$. Property (1) of the lemma follows for the base case from this observation.

Now, we prove the base case for property (2). Assume the condition of (2a) is satisfied, i.e., there are no cycle transitions in ρ. Now, we see that

$$\mathsf{Pr}[x_0, \rho'] = \int_{x_0}^{\infty} g'_0(x) \prod_{k \in S_1} \int_x^{\infty} g'_k(y)dy \prod_{k \in S_2} \int_{-\infty}^{x} g'_k(z)dz\, dx.$$

By introducing new variables u, v, w such $u = x - 1$, $v = y - 1$, $w = z - 1$, we get

$$\mathsf{Pr}[x_0, \rho'] = \int_{x_0-1}^{\infty} g'_0(u+1) \prod_{k \in S_1} \int_u^{\infty} g'_k(v + 1)dv \prod_{k \in S_2} \int_{-\infty}^{u} g'_k(w+1)dw\, du.$$

Observing that, for each $k \geq 0$, $g'_k(u+1) \geq e^{-2d_k\epsilon}g_k(u)$ and t_k is a critical transition, we get the inequality of property (2).

Now observe that condition of (2b) can not be satisfied as t_0 is the only assignment transition in ρ. Now, assume the condition of (2c) is satisfied. Now, observe that, for all $k \in S_1$, t_k is a critical transition. As before, we see that

$$\mathsf{Pr}[x_0, \rho'] = \int_{x_0}^{\infty} f(x) \prod_{k \in S_2} \int_{-\infty}^{x} g'_k(y)dy\, dx$$

where $f(x) = g'_0(x) \prod_{k \in S_1} \int_x^{\infty} g'_k(y)dy$. Now, using inequality (2) of Lemma 10, we see that

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0-1}^{\infty} f(x+1) \prod_{k \in S_2} \int_{-\infty}^{x} g_k(y)dy\, dx.$$

Now, observe that

$$f(x+1) = g_0(x - (\theta_0 - 1)) \prod_{k \in S_1} \int_{x+1}^{\infty} g_k(y - \theta_k)dy.$$

Introducing a new variable z and setting $z = y - 1$, we see that

$$f(x+1) = g_0(x - (\theta_0 - 1)) \prod_{k \in S_1} \int_x^\infty g_k(z - (\theta_k - 1))dz$$

and

$$f(x+1) \geq e^{-2(d_0 + \sum_{k \in S_1} d_k)\epsilon} g_0(x) \prod_{k \in S_1} \int_x^\infty g_k(z)dz.$$

From this and the above inequality, it is easily seen that

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-2(d_0 + \sum_{k \in S_2} d_k)\epsilon} \mathsf{Pr}[x_0 - 1, \rho].$$

From this we see that the inequality of property (2) holds.

The proof for the base case of Properties (3) and (4) is symmetric to those of properties (1) and (2) and is left out. To prove property (5) for the base case, we see that the proof is similar to those of properties (1) and (3) depending on whether G-cycle or L-cycle transitions appear. There are two minor differences. The first difference is that if the first transition is a non-input transition then $\theta_0 = 0$ and hence it only incurs a cost of d_0 and not $2d_0$. The second difference is that the lower limit of the outer integral will be $-\infty$ in the former case, while the upper limit of the outer integral being ∞ in the latter case. In either case, it is straightforward to see that property (5) holds.

Inductive Step: Now, we prove the inductive step as follows. Assume that all the properties hold when ρ has $\ell > 0$ assignments. Now, consider the case when ρ has $\ell + 1$ assignments. Let t_i, for $i > 0$, be the second assignment transition in ρ. Let S_1 (resp., S_2) be the set of k, $0 < k < i$, such that c_k is $\mathsf{insample} \geq \mathsf{x}$ (resp., $\mathsf{insample} < \mathsf{x}$).

Consider the case when c_0 is $\mathsf{insample} \geq \mathsf{x}$. Now, we consider two sub-cases. We first consider the sub-case when there is no cycle transitions before t_i. We have $\mathsf{Pr}[x_0, \rho'] = \int_{x_0}^\infty f'(x)\mathsf{Pr}[\rho'||i, x]dx$ where

$$f'(x) = g_0'(x) \prod_{k \in S_1} \int_x^\infty g_k'(y)dy \prod_{k \in S_2} \int_{-\infty}^x g_k'(y)dy.$$

Applying the inductive hypothesis for the suffix $\rho||i$, we get an inequality involving $\mathsf{Pr}[\rho'||i, x]$ and $\mathsf{Pr}[x+1, \rho||i]$, or $\mathsf{Pr}[\rho'||i, x-1]$, or $\mathsf{Pr}[x, \rho||i]$, based on which of the five properties of the lemma are satisfied by $\rho||i$. Suppose the condition of property (1) is satisfied by $\rho||i$, by using the inductive hypothesis, we get $\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^\infty f'(x)h(x)dx$, where $h(x) = e^{-2\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. Now, by taking $f(x) = f'(x)h(x)$, using inequality (1) of Lemma 10 and by taking $k = 0$ in that inequality, we get property (1) for the path ρ using the same simplification/reasoning used in the base case and by observing that

$$\mathsf{Pr}[x_0 + 1, \rho] = \int_{x_0+1}^\infty g_0(x) \prod_{k \in S_1} \int_x^\infty g_k(y)dy \prod_{k \in S_2} \int_{-\infty}^x g_k(y)dy\mathsf{Pr}[x, \rho||i]dx.$$

We can similarly prove the inductive step when the suffix $\rho||i$ satisfies the other properties (i.e., 2 through 5) of the lemma.

Now consider the sub-case when a cycle transition appears before t_i. Assume that the cycle transitions are G-cycle transitions. If c_i is also $\mathsf{insample} \geq \mathsf{x}$, then the suffix $\rho||i$ can satisfy any of the conditions of the first two properties of the lemma; In this situation, let $f(x) = f'(x)h(x)$ where $f'(x) = g_0'(x)\prod_{k \in S_2}\int_{-\infty}^x g_k'(y)dy$ and $h(x) = e^{-2\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. Observe that, if $\rho||i$ satisfies the condition of property (1) then $h(x)$ is the RHS of the inequality, we get, by applying the inductive hypothesis to $\rho||i$. If $\rho||i$ satisfies the condition of property (2) of the lemma then, by applying the inductive hypothesis to $\rho||i$, we get $\mathsf{Pr}[\rho'||i, x] \geq e^{-2\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x-1, \rho||i]$. Since, $\mathsf{Pr}[x-1, \rho||i] \geq \mathsf{Pr}[x+1, \rho||i]$, we see that $\mathsf{Pr}[\rho'||i, x] \geq e^{-2\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. Now, we have $\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^\infty f'(x)h(x)\prod_{k \in S_1}\int_x^\infty g_k'(z)dzdx$. Applying the inequality (1) of Lemma 10, we get the desired result for the inductive step. On the other hand, if c_i is $\mathsf{insample} < \mathsf{x}$ then the suffix $\rho||i$ can not satisfy the condition of property (3) of the lemma due to well-formedness of $\mathcal{A}$; however it can satisfy the condition of property (4). In this sub-case also, we can get the result for the induction case as above by using the inductive hypothesis for $\rho||i$ and using similar reasoning as in the base case and applying the first inequality of Lemma 10.

Now consider the situation where the cycle transitions appearing before t_i are L-cycle transitions. Now, we apply inequality (2) of Lemma 10 to prove that property (2) of the lemma is satisfied by ρ. To do this, we define $f(x) = f'(x)h(x)$ where $f'(x) = g_0'(x)\prod_{k \in S_1}\int_x^\infty g_k'(y)dy$ and $h(x) = e^{-2\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x-1, \rho||i]$. Next, applying the induction hypothesis to $\rho||i$, we show that

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^\infty f'(x)h(x) \prod_{k \in S_2} \int_{-\infty}^x g_k'(y)dydx.$$

Since $\mathcal{A}$ is well-formed, $\rho||i$ cannot satisfy the condition of property (1) of the lemma. If $\rho||i$ satisfies the condition of property (2) or that of property (3) then, the above inequality follows directly from the induction hypothesis; If $\rho||i$ satisfies the condition of property (4), then the above inequality follows from the induction hypothesis and the observation that $\mathsf{Pr}[x+1, \rho||i] \geq \mathsf{Pr}[x-1, \rho||i]$; If $\rho||i$ satisfies the condition of property (5) then the above inequality follows from the induction hypothesis and the observation that $\mathsf{Pr}[x, \rho||i] = \mathsf{Pr}[x-1, \rho||i]$ as $\mathsf{Pr}[x, \rho||i]$ is independent of x. Rewriting the above inequality, we get

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^\infty f'(x)h(x) \prod_{k \in S_2} \int_{-\infty}^x g_k(y - \theta_k)dy.$$

Now, using the inequality (2) of Lemma 10, and using simplifications and reasoning as in the base cases, we see that property (2) of the lemma is satisfied by ρ.

The proof for the inductive step for the case when c_0 is $\mathsf{insample} < \mathsf{x}$ is symmetric. For the case, when c_0 is true, the proof will be on the same lines excepting that if t_0 is a

non-input transition then it incurs a cost of d_0 only and the limits of the outer integrals are $-\infty$ and ∞. □

DiP automata with Finite and Infinite Outputs

We shall now show that if a DiPA $\mathcal{A}$ is well-formed then it is differentially private. For simplicity, we will assume that all states are input states. The case when the $\mathcal{A}$ includes non-input states can be dealt with similarly. Finally, we also assume that there are no transitions that output the value of $\mathsf{insample}'$. In case there are transitions from $\mathsf{insample}'$, Lemma 13 can be proved by appealing to the composition theorem of differential privacy (See Theorem 3.14 of [2].)

The following proposition follows directly from the definition of well-formed DiP automata.

Proposition 12. *Let $\mathcal{A}$ be a well-formed DiPA and ρ be a path of $\mathcal{A}$ starting from a reachable state. Then ρ satisfies the following properties.*

- *If ρ starts with an assignment transition t_0 and has no further assignment transitions, and has a* G*-cycle or an* L*-cycle transition then the output of t_0 is from Γ.*
- *If ρ has no assignment transitions and has a* G*-cycle (resp.,* L*-cycle) transition then the output of every transition in ρ, with guard* $\mathsf{insample} < \mathsf{x}$ *(resp.,* $\mathsf{insample} \geq \mathsf{x}$*), is from Γ.*
- *If ρ starts with an* L*-cycle (resp.,* G*-cycle) transition and is an* AG*-path (resp.,* AL*-path) then the output of every transition, with guard* $\mathsf{insample} \geq \mathsf{x}$ *(resp.,* $\mathsf{insample} < \mathsf{x}$*), is from Γ.*
- *If ρ is an* AG*-path (resp.,* AL*-path) ending with a* G*-cycle (resp.,* L*-cycle) then the output of every transition, with guard* $\mathsf{insample} < \mathsf{x}$ *(resp.,* $\mathsf{insample} \geq \mathsf{x}$*) , is from Γ.*

Please note that Lemma 3 is an immediate consequence of the following lemma.

Lemma 13. *Let $\mathcal{A} = (Q, \Sigma, \Gamma, q_{\mathsf{init}}, X, P, \delta)$ be a well-formed DiPA and ρ be a path of length $n > 0$ Let t_0 be the initial transition, i.e., the 0th transition of ρ, c_0 be its guard and o_0 be its output. Let t_0 be an assignment transition, and let ρ' be a path that is equivalent to ρ such that $\mathsf{inseq}(\rho')$ is a neighbor of $\mathsf{inseq}(\rho)$. Then the following properties hold for all $x_0 \in \mathbb{R}$.*

1) *If the guard c_0 is* $\mathsf{insample} \geq \mathsf{x}$*, and the first cycle transition in ρ is a* G*-cycle transition and no assignment transition with guard* $\mathsf{insample} < \mathsf{x}$ *appears before it, $o_0 \in \Gamma$ and*

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 + 1, \rho].$$

2) *If the guard c_0 is* $\mathsf{insample} \geq \mathsf{x}$ *and either, (a) ρ has no cycle transitions; or (b) the first cycle transition in ρ is a* G*-cycle transition and an assignment transition with guard* $\mathsf{insample} < \mathsf{x}$ *appears before it; or (c) the first cycle transition in ρ is an* L*-cycle transition, then*

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0, \rho].$$

Furthermore, if the output of every transition, whose guard is $\mathsf{insample} \geq \mathsf{x}$*, is from Γ, until the first assignment transition whose guard is* $\mathsf{insample} < \mathsf{x}$ *or until the end of ρ, then*

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 - 1, \rho].$$

3) *If the guard c_0 is* $\mathsf{insample} < x$ *and the first cycle transition in ρ is a* L*-cycle transition and no assignment transition with guard* $\mathsf{insample} \geq \mathsf{x}$ *appears before it, then $o_0 \in \Gamma$ and*

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 - 1, \rho].$$

4) *If the guard c_0 is* $\mathsf{insample} < x$*, and either (a) If ρ has no cycle transitions; or (b) The first cycle transition in ρ is an* L*-cycle transition and an assignment transition with guard* $\mathsf{insample} \geq \mathsf{x}$ *appears before it; or (c) the first cycle transition in ρ is a* G*-cycle transition, then*

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0, \rho].$$

Furthermore, if the output of every transition, whose guard is $\mathsf{insample} < \mathsf{x}$*, is from Γ, until the first assignment transition whose guard is* $\mathsf{insample} \geq \mathsf{x}$ *or until the end of ρ, then*

$$\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0 + 1, \rho].$$

5) *If the guard c_0 is* true*, then* $\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon}\mathsf{Pr}[x_0, \rho]$.

Proof. Let $\rho = q_0 \xrightarrow{a_0,o_0} q_1 \xrightarrow{a_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{a_{n-1},o_{n-1}} q_n$ and $\rho' = q_0 \xrightarrow{b_0,o_0} q_1 \xrightarrow{b_1,o_1} q_2 \cdots q_{n-1} \xrightarrow{b_{n-1},o_{n-1}} q_n$. Let $t_0, \ldots, t_{n-1}$ be the transitions of ρ and let $c_0, \ldots, c_{n-1}$ be their respective guards. For each $k \leq n$, let d_k, μ_k be such that $P(q_k) = (d_k, \mu_k)$. Recall that, for any k, $\rho||k$ denotes the suffix of ρ starting from q_k. Once again, we assume that there are no cycle transitions that are assignments.

We show, how the proof of Lemma 11 can be modified to prove this Lemma. First, observe that properties (1), (3) and (5) of the Lemma are identical to the corresponding properties of the Lemma 11. When $o_i \in \Gamma$, for all $i, 0 \leq i < n$, the second parts of the properties (2) and (4) subsume their first parts, and these two properties become identical to properties (2) and (4) of the Lemma 11, respectively. For each $i, 0 \leq i < n$, let (u_i, v_i) be such that $o_i = (\mathsf{insample}, u_i, v_i)$ if $o_i \notin \Gamma$, otherwise it is the interval $(-\infty, \infty)$. Let $g_k(y), g'_k(y)$ be the functions as defined in the proof of Lemma 11, and $\theta_k = b_k - a_k$ for $0 \leq k < n$.

As before, we prove the Lemma by induction on the number of assignment transitions in ρ. In the base case, ρ has one assignment transition which is t_0. Let S_1 and S_2 be the sets of $k > 0$ such that c_k is $\mathsf{insample} \geq x$ and c_k is $\mathsf{insample} < x$, respectively.

Now, assume the condition of (1) is satisfied. Observe that S_1 includes all G-cycle transitions whose guard is $\mathsf{insample} \geq \mathsf{x}$. Let S'_1 be the set of $k \in S_1$ such that t_k is a G-cycle transition and $S''_1 = S_1 \setminus S'_1$. Observe that, using the fact that $\mathcal{A}$ is well-formed and using Proposition 12, we see the following hold: (i) for all $k \in S'_1 \cup S_2$, $o_k \in \Gamma$; (ii) t_0 is a critical transition and $o_0 \in \Gamma$; (iii) for all $k \in S_2 \cup S''_1$, t_k does

not lie on a cycle and hence is a critical transition. Note that, for any $k \in S_1''$, o_k may be insample. Now, we see that

$$\mathsf{Pr}[x_0, \rho'] \;=\; \int_{x_0}^{\infty} f(x) \prod_{k \in S_1'} \int_{x}^{\infty} g_k'(y) dy \, dx$$

where

$$f(x) \;=\; g_0'(x) \prod_{k \in S_2} \int_{-\infty}^{x} g_k'(y) dy \prod_{k \in S_1''} \int_{\max(x,u_k)}^{v_k} g_k'(z) dz.$$

Now, substituting $g_k'(y) = g_k(y - \theta_k)$ (for $k \in S_1$) in the above equation and using inequality (1) of Lemma 10, we see that

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0+1}^{\infty} f(x-1) \prod_{k \in S_1} \int_{x}^{\infty} g_k(y) dy \, dx.$$

Now, using the same argument as in the proof of Lemma 11, and observing that, for $k \in S_1''$, $\int_{\max(x-1,u_k)}^{v_k} g_k'(z) dz \geq \int_{\max(x,u_k)}^{v_k} g_k'(z) dz$, it is easy to see that

$$\begin{aligned} f(x-1) \quad \geq \quad & e^{-2(d_{q_0} + \sum_{k \in S_1'' \cup S_2} d_{q_k})\epsilon} g_0(x) \\ & \prod_{k \in S_2} \int_{-\infty}^{x} g_k(y) dy \\ & \prod_{k \in S_1''} \int_{\max(x,u_k)}^{v_k} g_k'(z) dz. \end{aligned}$$

Putting all the above observations together, we see that property (1) holds.

Now, we prove the base case for property (2). Assume the condition of (2a) is satisfied, i.e., there are no cycle transitions in ρ. Now, we see that

$$\begin{aligned} \mathsf{Pr}[x_0, \rho'] \quad = \quad & \int_{\max(x_0,u_0)}^{v_0} g_0'(x) \prod_{k \in S_1} \int_{\max(x,u_k)}^{v_k} g_k'(y) dy \\ & \prod_{k \in S_2} \int_{u_k}^{\min(x,v_k)} g_k'(z) dz \, dx. \end{aligned}$$

It is fairly straightforward to see that $\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon} \mathsf{Pr}[x_0, \rho]$ since $g_k'(y) \geq e^{-d_{q_k}\epsilon} g_k(y)$, for all $y \in \mathbb{R}$, $0 \leq k < n$. From this, we see that the first part of property(2) holds. To see that the second part of property (2) holds, assume that $o_0 \in \Gamma$, and for all $k \in S_1, o_k \in \Gamma$. This means that

$$\begin{aligned} & \mathsf{Pr}[x_0, \rho'] \\ & = \int_{x_0}^{\infty} g_0'(x) \prod_{k \in S_1} \int_{x}^{\infty} g_k'(y) dy \prod_{k \in S_2} \int_{u_k}^{\min(x,v_k)} g_k'(z) dz dx. \end{aligned}$$

Now introducing new variables w, y' and setting $w = x - 1$ and $y' = y - 1$, we see that

$$\begin{aligned} \mathsf{Pr}[x_0, \rho'] \quad = \quad & \int_{x_0-1}^{\infty} g_0'(w+1) \prod_{k \in S_1} \int_{w}^{\infty} g_k'(y'+1) dy' \\ & \prod_{k \in S_2} \int_{u_k}^{\min(w+1,v_k)} g_k'(z) dz \, dx. \end{aligned}$$

Now, observe that, for $k \in S_2$, $\int_{u_k}^{\min(w+1,v_k)} g_k'(z) dz \geq \int_{u_k}^{\min(w,v_k)} g_k'(z) dz$. Using this we get,

$$\begin{aligned} \mathsf{Pr}[x_0, \rho'] \quad \geq \quad & \int_{x_0-1}^{\infty} g_0'(w+1) \prod_{k \in S_1} \int_{w}^{\infty} g_k'(y'+1) dy' \\ & \prod_{k \in S_2} \int_{u_k}^{\min(w,v_k)} g_k'(z) dz \, dx. \end{aligned}$$

Now, the second part of property (2), follows from the above inequality and the reasoning employed earlier.

Now, condition of (2b) can not be satisfied as t_0 is the only assignment transition in ρ. Now, assume the condition of (2c) is satisfied. Let S_2' be the set of all $k \in S_2$ such that t_k is an L-cycle transition and $S_2'' = S_2 \setminus S_2'$. Now, using the fact that $\mathcal{A}$ is well-formed and using Proposition 12 we observe that the following hold: (i) for all $k \in S_1 \cup S_2''$, t_k is a critical transition; (ii) t_0 is a critical transition and $o_0 \in \Gamma$; (iii) for all $k \in S_1 \cup S_2'$, $o_k \in \Gamma$. Now, we see that that

$$\mathsf{Pr}[x_0, \rho'] \;=\; \int_{x_0}^{\infty} f(x) \prod_{k \in S_2'} \int_{-\infty}^{x} g_k'(y) dy \, dx$$

where

$$f(x) \;=\; g_0'(x) \prod_{k \in S_1} \int_{x}^{\infty} g_k'(y) dy \prod_{k \in S_2''} \int_{u_k}^{\min(x,v_k)} g_k'(y) dy.$$

Now, using inequality (2) of Lemma 10, we see that

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0-1}^{\infty} f(x+1) \prod_{k \in S_2'} \int_{-\infty}^{x} g_k(y) dy \, dx.$$

Now, observe that

$$\begin{aligned} f(x+1) \quad = \quad & g_0(x - (\theta_0 - 1)) \prod_{k \in S_1} \int_{x+1}^{\infty} g_k(y - \theta_k) dy \\ & \prod_{k \in S_2''} \int_{u_k}^{\min(x+1,v_k)} g_k'(y) dy. \end{aligned}$$

Introducing a new variable z and setting $z = y - 1$, we see that

$$\begin{aligned} f(x+1) \quad = \quad & g_0(x - (\theta_0 - 1)) \prod_{k \in S_1} \int_{x}^{\infty} g_k(z - (\theta_k - 1)) dz \\ & \prod_{k \in S_2''} \int_{u_k}^{\min(x+1,v_k)} g_k'(y) dy \end{aligned}$$

and

$$\begin{aligned} f(x+1) \quad \geq \quad & e^{-2(d_{q_0} + \sum_{k \in S_1 \cup S_2''} d_{q_k})\epsilon} g_0(x) \\ & \prod_{k \in S_1} \int_{x}^{\infty} g_k(z) dz \prod_{k \in S_2''} \int_{u_k}^{\min(x,v_k)} g_k(y). \end{aligned}$$

From this and the above inequality, it is easily seen that $\mathsf{Pr}[x_0, \rho'] \geq e^{-\mathsf{wt}(\rho)\epsilon} \mathsf{Pr}[x_0 - 1, \rho]$. From this we see that the inequalities of both parts of property (2) hold.

As before, the proof for the base case of Properties (3) and (4) is symmetric to those of properties (1) and (2) and is left out. Property (5) is proved as in the case of Lemma 11.

Now, we prove the inductive step as follows. Assume that all the properties hold when ρ has $\ell > 0$ assignments. Now, consider the case when ρ has $\ell + 1$ assignments. Let t_i, for $i > 0$, be the second assignment transition in ρ. Let S_1 (resp., S_2) be the set of k, $0 < k < i$, such that c_k is $\mathsf{insample} \geq x$ (resp., $\mathsf{insample} < \mathsf{x}$). Now, consider the case when c_0 is $\mathsf{insample} \geq \mathsf{x}$. Now, we consider two sub-cases. We first consider the sub-case when there is no cycle transitions before t_i. We have $\mathsf{Pr}[x_0, \rho'] = \int_{\max(x_0,u_0)}^{v_0} f'(x)\mathsf{Pr}[x, \rho'||i]dx$ where $f'(x) = g'_0(x)\prod_{k\in S_1}\int_{\max(x,u_k)}^{v_k} g'_k(y)dy \prod_{k\in S_2}\int_{u_k}^{\min(x,v_k)} g'_k(y)dy$. Applying the inductive hypothesis for the suffix $\rho||i$, we get an inequality involving $\mathsf{Pr}[x, \rho'||i]$ and $\mathsf{Pr}[x+1, \rho||i]$, or $\mathsf{Pr}[x-1, \rho||i]$, or $\mathsf{Pr}[x, \rho||i]$, based on which of the five properties of the Lemma are satisfied by $\rho||i$. Suppose the condition of property (1) is satisfied by $\rho||i$. Let $j \geq i$ be the smallest integer such that t_j is a G-cycle transition. Now, since $p|j$, the prefix of ρ, is an AG-path, using the fact that $\mathcal{A}$ is well-formed and using Proposition 12, it is easy to see that $o_0 \in \Gamma$, and for all $k \in S_2$, $o_k \in \Gamma$. By using the inductive hypothesis, we get $\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^{\infty} f'(x)h(x)dx$, where $h(x) = e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. Because of the previous observation, we see that $f'(x) = g'_0(x)\prod_{k\in S_1}\int_{\max(x,u_k)}^{v_k} g'_k(y)dy \prod_{k\in S_2}\int_{-\infty}^{x} g'_k(y)dy$. Now, observe that, for each $k \in S_1$, $\int_{\max(x-1,u_k)}^{v_k} g'_k(y)dy \geq \int_{\max(x,u_k)}^{v_k} g'_k(y)dy$. From this, using the reasoning employed in the base case, we see that

$$f'(x-1) \geq e^{-2(d_{q_0}+\sum_{k\in S_1\cup S_2} d_{q_k})\epsilon} g_0(x) \prod_{k\in S_1}\int_{\max(x,u_k)}^{v_k} g_k(y)dy \prod_{k\in S_2}\int_{-\infty}^{x} g_k(y)dy.$$

Now, by taking $f(x) = f'(x)h(x)$, using inequality (1) of Lemma 10 and by taking $k = 0$ in that inequality, we get property (1) for the path ρ using the same simplification/reasoning used in the base case and by observing that

$$G_p(x_0+1) = \int_{x_0+1}^{\infty} g_0(x) \prod_{k\in S_1}\int_{\max(x,u_k)}^{v_k} g_k(y)dy \prod_{k\in S_2}\int_{-\infty}^{x} g_k(y)dy\mathsf{Pr}[x, \rho||i]dx.$$

We can similarly prove the inductive step when the suffix $\rho||i$ satisfies the other properties (i.e., 2 through 5) of the Lemma.

Now consider the sub-case when a cycle transition appears before t_i. Assume that the cycle transitions are G-cycle transitions. Let S'_1 be the set of $k \in S_1$ such that t_k is a G-cycle transition and $S''_1 = S_1 \setminus S'_1$. Since $\mathcal{A}$ is well-formed, using Proposition 12, we see that $o_0 \in \Gamma$, and for every $k \in S'_1 \cup S_2$, $o_k \in \Gamma$. Let $f(x) = f'(x)h(x)$ where $f'(x) = g'_0(x)\prod_{k\in S_2}\int_{-\infty}^{x} g'_k(y)dy \prod_{k\in S''_1}\int_{\max(x,u_k)}^{v_k} g'_k(y)dy$ and $h(x) = e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. If c_i is also $\mathsf{insample} \geq \mathsf{x}$, then the suffix $\rho||i$ can satisfy any of the conditions of the first two properties of the Lemma; In this situation, observe that, if $\rho||i$ satisfies the condition of property (1) then $h(x)$ is the right handside of the inequality, we get, by applying the inductive hypothesis to $\rho||i$; If $\rho||i$ satisfies the condition of property (2) of the Lemma then, by applying the inductive hypothesis to $\rho||i$, we get $\mathsf{Pr}[x, \rho'||i] \geq e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x, \rho||i]$; since, $\mathsf{Pr}[x, \rho||i] \geq \mathsf{Pr}[x+1, \rho||i]$, we see that $\mathsf{Pr}[x, \rho'||i] \geq e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. Now, assume that c_i is $\mathsf{insample} < \mathsf{x}$. Now, since $\mathcal{A}$ is well-formed, it is easy to see that the condition of property (3) of the Lemma cannot be satisfied. Assume that $\rho||i$ satisfies the condition of property (4) of the Lemma. Let k' be the smallest integer such that, $i \leq k' \leq n$, and either $k' = n$, or $t_{k'}$ is an assignment transition and $c_{k'}$ is $\mathsf{insample} \geq \mathsf{x}$. Now, we see that the path starting with t_1 and ending with $t_{k'-1}$ is an AL-path. Using Proposition 12 and the fact that $\mathcal{A}$ is well-formed, we see that, for all $j, i \leq j < k'$, such that c_j is $\mathsf{insample} < \mathsf{x}$, $o_j \in \Gamma$. Now, applying the induction hypothesis for $\rho||i$, using the second part of property (4), we get $\mathsf{Pr}[x, \rho'||i] \geq e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. Now, if c_i is true, applying the induction hypothesis and using property (5), we see that $\mathsf{Pr}[x, \rho'||i] \geq e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x, \rho||i]$; since $\mathsf{Pr}[x, \rho||i]$ is independent of x, we see that $\mathsf{Pr}[x, \rho'||i] \geq e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x+1, \rho||i]$. Thus, irrespective of what guard c_i is, we have $\mathsf{Pr}[x, \rho'||i] \geq h(x)$. Now, we have $\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^{\infty} f'(x)h(x)\prod_{k\in S'_1}\int_{x}^{\infty} g'_k(z)dzdx$. Applying the inequality (1) of Lemma 10, we get $\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0+1}^{\infty} f'(x-1)h(x-1)\prod_{k\in S'_1}\int_{x}^{\infty} g_k(z)dzdx$. Observe that, for $k \in S''_1$, $\int_{\max(x-1,u_k)}^{v_k} g'_k(y)dy \geq \int_{\max(x,u_k)}^{v_k} g'_k(y)dy$. Using this observation and the reasoning/simplification as in the base case, we see that property (1) is satisfied by ρ.

Now consider the situation where the cycle transitions appearing before t_i are L-cycle transitions. Now, we apply inequality (2) of Lemma 10 to prove that property (2) of the Lemma is satisfied by ρ. Let S'_2 be the set of $k \in S_2$ such that t_k is an L-cycle transition and $S''_2 = S_2 \setminus S'_2$. Since $\mathcal{A}$ is well-formed, using Proposition 12, we see that $o_0 \in \Gamma$, and for every $k \in S_1 \cup S'_2$, $o_k \in \Gamma$. Now, let $f(x) = f'(x)h(x)$ where $f'(x) = g'_0(x)\prod_{k\in S_1}\int_{x}^{\infty} g'_k(y)dy \prod_{k\in S''_2}\int_{u_k}^{\min(x,v_k)} g'_k(z)dz$ and $h(x) = e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x-1, \rho||i]$. Now, applying the induction hypothesis to $\rho||i$, we show that

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^{\infty} f'(x)h(x) \prod_{k\in S'_2}\int_{-\infty}^{x} g'_k(y)dydx.$$

Since $\mathcal{A}$ is well-formed $\rho||i$ cannot satisfy the condition of property (1). Now, consider the case when $\rho||i$ satisfies the condition of property (2). Let k' be the smallest integer such that, $i \leq k' \leq n$, and either $k' = n$ or $t_{k'}$ is an assignment transition and $c_{k'}$ is $\mathsf{insample} < \mathsf{x}$. Now, we see that the path starting with t_i and ending with $t_{k'-1}$ is a AG-path. From this observation, using the fact that $\mathcal{A}$ is well-formed and using Proposition 12, we see that, for all $j, i \leq j < k'$, such that c_j is $\mathsf{insample} \geq \mathsf{x}$, $o_j \in \Gamma$. Now, applying the induction hypothesis for $\rho||i$, using the second part of property (2), we get $\mathsf{Pr}[x, \rho'||i] \geq e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x-1, \rho||i]$.

If $\rho||i$ satisfies property (3), then we directly see from the induction hypothesis $\mathsf{Pr}[x, \rho'||i] \geq e^{-\mathsf{wt}(\rho||i)\epsilon}\mathsf{Pr}[x-1, \rho||i]$. If $\rho||i$ satisfies property(4), we get the above inequality, using the first part of the induction hypothesis and the observation that $\mathsf{Pr}[x, \rho||i] \geq \mathsf{Pr}[x-1, \rho||i]$. If $\rho||i$ satisfies property (5) then, we get the above inequality from the induction hypothesis and the observation that $\mathsf{Pr}[x, \rho||i]$ is independent of x. In all the above cases, it is easy to see,

$$\mathsf{Pr}[x_0, \rho'] \geq \int_{x_0}^{\infty} f'(x)h(x) \prod_{k \in S_2'} \int_{-\infty}^{x} g_k(y - \theta_k)dy.$$

Now, using the inequality (2) of Lemma 10, and observing that, for all $k \in S_2''$, $\int_{u_k}^{\min(x+1, v_k)} g_k'(z)dz \geq \int_{u_k}^{\min(x, v_k)} g_k'(z)dz$, and using simplifications and reasoning as in the base cases, we see that property (2) of the Lemma is satisfied by ρ.

□