

# Differential Privacy using Quasi-Borel Spaces

Michikazu Hirata

February 6, 2026

## Abstract

This entry formalizes differential privacy using quasi-Borel spaces. In general, differential privacy is discussed using measurable spaces. Sato and Katsumata showed that quasi-Borel spaces are also applied to formulate differential privacy [1]. We formalize basic definitions and properties of differential privacy using quasi-Borel spaces, and show two examples: randomized response and the naive report noisy max algorithm.

## Contents

<b>1</b>	<b>Definitions</b>	<b>1</b>
1.1	Divergence for Differential Privacy using QBS . . . . .	2
1.2	Differential Privacy using QBS . . . . .	4
<b>2</b>	<b>Examples</b>	<b>5</b>
2.1	Randomized Response . . . . .	6
2.2	Laplace Distribution in QBS . . . . .	6
2.3	Naive Report Noisy Max Mechanism . . . . .	6

**theory** *DP-QBS*

**imports** *Differential-Privacy.Differential-Privacy-Divergence*  
*Differential-Privacy.Differential-Privacy-Standard*  
*S-Finite-Measure-Monad.Monad-QuasiBorel*

**begin**

**declare** *qbs-morphism-imp-measurable[measurable-dest]*

## 1 Definitions

Details of differential privacy using quasi-Borel spaces are found at [1]

## 1.1 Divergence for Differential Privacy using QBS

**definition** *DP-qbs-divergence* :: 'a qbs-measure  $\Rightarrow$  'a qbs-measure  $\Rightarrow$  real  $\Rightarrow$  ereal  
*(DP'-divergence<sub>Q</sub>)* **where**  
*DP-qbs-divergence-qbs-l*: *DP-divergence<sub>Q</sub>* p q e  $\equiv$  *DP-divergence* (qbs-l p) (qbs-l q)  
e

**abbreviation** *DP-qbs-inequality* (*DP'-inequality<sub>Q</sub>*) **where**  
*DP-qbs-inequality* p q  $\varepsilon$   $\delta$   $\equiv$  *DP-divergence<sub>Q</sub>* p q  $\varepsilon$   $\leq$  ereal  $\delta$

**lemmas** *DP-qbs-divergence-def* = *DP-qbs-divergence-qbs-l*[*simplified DP-divergence-SUP*]

**lemma** *DP-qbs-divergence-nonneg[simp]*:  $0 \leq$  *DP-divergence<sub>Q</sub>* p q e  
⟨*proof*⟩

**lemma** *DP-qbs-divergence-le-ereal-iff*:  
*DP-divergence<sub>Q</sub>* p q  $\varepsilon$   $\leq$  ereal  $\delta$   $\iff$  ( $\forall A \in$  sets (qbs-l p). *measure* (qbs-l p) A -  
exp  $\varepsilon$  \* *measure* (qbs-l q) A  $\leq$   $\delta$ )  
⟨*proof*⟩

**corollary** *DP-qbs-divergence-le-ereal-dest*:  
**assumes** *DP-divergence<sub>Q</sub>* p q  $\varepsilon$   $\leq$  ereal  $\delta$   
**shows** *measure* (qbs-l p) A  $\leq$  exp  $\varepsilon$  \* *measure* (qbs-l q) A +  $\delta$   
⟨*proof*⟩

**corollary** *DP-qbs-divergence-le-erealI*:  
**assumes**  $\bigwedge A. A \in$  sets (qbs-l p)  $\implies$  *measure* (qbs-l p) A  $\leq$  exp  $\varepsilon$  \* *measure*  
(qbs-l q) A +  $\delta$   
**shows** *DP-divergence<sub>Q</sub>* p q  $\varepsilon$   $\leq$  ereal  $\delta$   
⟨*proof*⟩

**lemma** *DP-qbs-divergence-zero*:  
**assumes** p  $\in$  monadP-qbs X  
**and** q  $\in$  monadP-qbs X  
**and** *DP-inequality<sub>Q</sub>* p q 0 0  
**shows** p = q  
⟨*proof*⟩

**lemma** *DP-qbs-divergence-antimono*: a  $\leq$  b  $\implies$  *DP-divergence<sub>Q</sub>* p q b  $\leq$  *DP-divergence<sub>Q</sub>*  
p q a  
⟨*proof*⟩

**lemma** *DP-qbs-divergence-refl[simp]*: *DP-divergence<sub>Q</sub>* p p 0 = 0  
⟨*proof*⟩

**lemma** *DP-qbs-divergence-refl'[simp]*:  $0 \leq$  e  $\implies$  *DP-divergence<sub>Q</sub>* p p e = 0  
⟨*proof*⟩

**lemma** *DP-qbs-divergence-trans'*:  
**assumes** *DP-inequality<sub>Q</sub>*  $p\ q\ \varepsilon\ \delta$   
**and** *DP-inequality<sub>Q</sub>*  $q\ l\ \varepsilon'\ 0$   
**shows** *DP-inequality<sub>Q</sub>*  $p\ l\ (\varepsilon + \varepsilon')\ \delta$   
 $\langle$ *proof* $\rangle$

**lemmas** *DP-qbs-divergence-trans* = *DP-qbs-divergence-trans'*[**where**  $\delta=0$ ]

**proposition** *DP-qbs-divergence-compose*:  
**assumes** [*qbs,measurable*]: $p \in \text{monadP-qbs } X\ q \in \text{monadP-qbs } X\ f \in X \rightarrow_Q \text{monadP-qbs } Y\ g \in X \rightarrow_Q \text{monadP-qbs } Y$   
**and** *dp1*:*DP-divergence<sub>Q</sub>*  $p\ q\ \varepsilon \leq \text{ereal } \delta$   
**and** *dp2*: $\bigwedge x. x \in \text{qbs-space } X \implies \text{DP-divergence}_Q (f\ x)\ (g\ x)\ \varepsilon' \leq \text{ereal } \delta'$   
**and** [*arith*]: $0 \leq \varepsilon\ 0 \leq \varepsilon'$   
**shows** *DP-divergence<sub>Q</sub>*  $(p \ggg f)\ (q \ggg g)\ (\varepsilon + \varepsilon') \leq \text{ereal } (\delta + \delta')$   
 $\langle$ *proof* $\rangle$

**corollary** *DP-qbs-divergence-dataprocessing*:  
**assumes** [*qbs*]: $p \in \text{monadP-qbs } X\ q \in \text{monadP-qbs } X\ f \in X \rightarrow_Q \text{monadP-qbs } Y$   
**and** *dp*: *DP-divergence<sub>Q</sub>*  $p\ q\ \varepsilon \leq \text{ereal } \delta$   
**and** [*arith*]: $0 \leq \varepsilon$   
**shows** *DP-divergence<sub>Q</sub>*  $(p \ggg f)\ (q \ggg f)\ \varepsilon \leq \text{ereal } \delta$   
 $\langle$ *proof* $\rangle$

**lemma** *DP-qbs-divergence-additive*:  
**assumes** [*qbs*]: $p \in \text{monadP-qbs } X\ q \in \text{monadP-qbs } X\ p' \in \text{monadP-qbs } Y\ q' \in \text{monadP-qbs } Y$   
**and** *div1*: *DP-divergence<sub>Q</sub>*  $p\ q\ \varepsilon \leq \text{ereal } \delta$   
**and** *div2*: *DP-divergence<sub>Q</sub>*  $p'\ q'\ \varepsilon' \leq \text{ereal } \delta'$   
**and** [*arith*]: $0 \leq \varepsilon\ 0 \leq \varepsilon'$   
**shows** *DP-divergence<sub>Q</sub>*  $(p \otimes_{Q\text{mes}} p')\ (q \otimes_{Q\text{mes}} q')\ (\varepsilon + \varepsilon') \leq \text{ereal } (\delta + \delta')$   
 $\langle$ *proof* $\rangle$

**corollary** *DP-qbs-divergence-strength*:  
**assumes** [*qbs*]: $p \in \text{monadP-qbs } X\ q \in \text{monadP-qbs } X\ x \in \text{qbs-space } Y$   
**and** *dp*: *DP-divergence<sub>Q</sub>*  $p\ q\ \varepsilon \leq \text{ereal } \delta$   
**and** [*simp*]: $0 \leq \varepsilon$   
**shows** *DP-divergence<sub>Q</sub>*  $(\text{return-qbs } Y\ x \otimes_{Q\text{mes}} p)\ (\text{return-qbs } Y\ x \otimes_{Q\text{mes}} q)$   
 $\varepsilon \leq \text{ereal } \delta$   
 $\langle$ *proof* $\rangle$

## 1.2 Differential Privacy using QBS

**definition** *DP-qbs* (*differential'-privacy<sub>Q</sub>*) **where**

*DP-qbs-qbs-L:differential-privacy<sub>Q</sub>*  $M \equiv \text{differential-privacy } (\lambda x. \text{qbs-l } (M x))$

**lemma** *DP-qbs-def*:

*differential-privacy<sub>Q</sub>*  $M \text{ adj } \varepsilon \delta \longleftrightarrow$   
 $(\forall (d1, d2) \in \text{adj}. \text{DP-inequality}_Q (M d1) (M d2) \varepsilon \delta \wedge \text{DP-inequality}_Q (M d2) (M d1) \varepsilon \delta)$   
*<proof>*

**lemma** *DP-qbs-adj-sym*:

**assumes** *sym adj*  
**shows** *differential-privacy<sub>Q</sub>*  $M \text{ adj } \varepsilon \delta \longleftrightarrow (\forall (d1, d2) \in \text{adj}. \text{DP-inequality}_Q (M d1) (M d2) \varepsilon \delta)$   
*<proof>*

**lemma** *pure-DP-qbs-comp*:

**assumes**  $\text{adj} \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
**and**  $\text{adj}' \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
**and** *differential-privacy<sub>Q</sub>*  $M \text{ adj } \varepsilon 0$   
**and** *differential-privacy<sub>Q</sub>*  $M \text{ adj}' \varepsilon' 0$   
**and**  $M \in X \rightarrow_Q \text{monadP-qbs } Y$   
**shows** *differential-privacy<sub>Q</sub>*  $M (\text{adj } O \text{adj}') (\varepsilon + \varepsilon') 0$   
*<proof>*

**lemma** *pure-DP-qbs-trans-k*:

**assumes**  $\text{adj} \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
**and** *differential-privacy<sub>Q</sub>*  $M \text{ adj } \varepsilon 0$   
**and**  $M \in X \rightarrow_Q \text{monadP-qbs } Y$   
**shows** *differential-privacy<sub>Q</sub>*  $M (\text{adj} \overset{\sim}{\sim} k) (k * \varepsilon) 0$   
*<proof>*

**proposition** *DP-qbs-postprocessing*:

**assumes**  $\varepsilon \geq 0$   
**and** *differential-privacy<sub>Q</sub>*  $M \text{ adj } \varepsilon \delta$   
**and**  $[\text{qbs,measurable}]: M \in X \rightarrow_Q \text{monadP-qbs } Y$   
**and**  $[\text{qbs,measurable}]: N \in Y \rightarrow_Q \text{monadP-qbs } Z$   
**and**  $\text{adj} \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
**shows** *differential-privacy<sub>Q</sub>*  $(\lambda x. M x \gg N) \text{ adj } \varepsilon \delta$   
*<proof>*

**corollary** *DP-qbs-postprocessing-return*:

**assumes**  $\varepsilon \geq 0$   
**and** *differential-privacy<sub>Q</sub>*  $M \text{ adj } \varepsilon \delta$   
**and**  $M \in X \rightarrow_Q \text{monadP-qbs } Y$   
**and**  $N \in Y \rightarrow_Q Z$   
**and**  $\text{adj} \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
**shows** *differential-privacy<sub>Q</sub>*  $(\lambda x. M x \gg (\lambda y. \text{return-qbs } Z (N y))) \text{ adj } \varepsilon \delta$   
*<proof>*

**lemma** *DP-qbs-preprocessing*:

assumes  $\varepsilon \geq 0$   
 and *differential-privacy*<sub>Q</sub>  $M$  *adj*  $\varepsilon$   $\delta$   
 and *[measurable]*:  $f \in X' \rightarrow_Q X$   
 and  $\forall (x,y) \in \text{adj}' . ((f\ x), (f\ y)) \in \text{adj}$   
 and  $\text{adj} \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
 and  $\text{adj}' \subseteq \text{qbs-space } X' \times \text{qbs-space } X'$   
 shows *differential-privacy*<sub>Q</sub>  $(M \circ f)$  *adj'*  $\varepsilon$   $\delta$   
*<proof>*

**proposition** *DP-qbs-bind-adaptive*:

assumes  $\varepsilon \geq 0$  and  $\varepsilon' \geq 0$   
 and *[qbs]*:  $M \in X \rightarrow_Q \text{monadP-qbs } Y$   
 and *differential-privacy*<sub>Q</sub>  $M$  *adj*  $\varepsilon$   $\delta$   
 and *[qbs]*:  $N \in X \Rightarrow_Q Y \Rightarrow_Q \text{monadP-qbs } Z$   
 and  $\bigwedge y . y \in \text{qbs-space } Y \implies \text{differential-privacy}_Q (\lambda x . N\ x\ y)$  *adj*  $\varepsilon'$   $\delta'$   
 and  $\text{adj} \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
 shows *differential-privacy*<sub>Q</sub>  $(\lambda x . M\ x \gg N\ x)$  *adj*  $(\varepsilon + \varepsilon')$   $(\delta + \delta')$   
*<proof>*

**proposition** *DP-qbs-bind-pair*:

assumes  $\varepsilon \geq 0$   $\varepsilon' \geq 0$   
 and *[qbs]*:  $M \in X \rightarrow_Q \text{monadP-qbs } Y$   
 and *differential-privacy*<sub>Q</sub>  $M$  *adj*  $\varepsilon$   $\delta$   
 and *[qbs]*:  $N \in X \rightarrow_Q \text{monadP-qbs } Z$   
 and *differential-privacy*<sub>Q</sub>  $N$  *adj*  $\varepsilon'$   $\delta'$   
 and  $\text{adj} \subseteq \text{qbs-space } X \times \text{qbs-space } X$   
 shows *differential-privacy*<sub>Q</sub>  $(\lambda x . M\ x \gg (\lambda y . N\ x \gg (\lambda z . \text{return-qbs } (Y \otimes_Q Z) (y,z))))$  *adj*  $(\varepsilon + \varepsilon')$   $(\delta + \delta')$   
*<proof>*

end

## 2 Examples

**theory** *DP-QBS-Examples*

imports *DP-QBS*

*Differential-Privacy.Differential-Privacy-Randomized-Response*

begin

**lemma** *qbs-space-list-qbs-borel*[*qbs*]:  $\bigwedge r . r \in \text{qbs-space } (\text{list-qbs borel}_Q)$

and *qbs-space-list-qbs-count-space*[*qbs*]:  $\bigwedge i . r \in \text{qbs-space } (\text{list-qbs } (\text{count-space}_Q (UNIV :: - :: \text{countable})))$

*<proof>*

## 2.1 Randomized Response

**lemma** *qbs-morphism-RR-mechanism*[qbs]: *qbs-pmf*  $\circ$  *RR-mechanism*  $e \in \text{count-space}_Q$   
 $UNIV \rightarrow_Q \text{monadP-qbs } (\text{count-space}_Q UNIV)$   
 ⟨proof⟩

**lemma** *qbs-DP-RR-mechanism*:

**assumes** [arith]:  $\varepsilon \geq 0$

**shows** *DP-divergence* $_Q$  (*RR-mechanism*  $\varepsilon$   $x$ ) (*RR-mechanism*  $\varepsilon$   $y$ )  $\varepsilon = 0$

⟨proof⟩

## 2.2 Laplace Distribution in QBS

**lemma** *qbs-morphism-laplace-density*[qbs]: *laplace-density*  $\in \text{borel}_Q \Rightarrow_Q \text{borel}_Q \Rightarrow_Q$   
 $\text{borel}_Q \Rightarrow_Q \text{borel}_Q$   
 ⟨proof⟩

**definition** *qbs-Lap-mechanism* (*Lap'-mechanism* $_Q$ ) **where**

*Lap-mechanism* $_Q \equiv \lambda e x. \text{if } e \leq 0 \text{ then return-qbs borel}_Q x \text{ else density-qbs lborel}_Q$   
 (*laplace-density*  $e$   $x$ )

**lemma** *qbs-morphism-Lap-mechanism*[qbs]: *Lap-mechanism* $_Q \in \text{borel}_Q \rightarrow_Q \text{borel}_Q$   
 $\Rightarrow_Q \text{monadP-qbs borel}_Q$   
 ⟨proof⟩

**lemma** *qbs-l-Lap-mechanism*: *qbs-l* (*Lap-mechanism* $_Q$   $e$   $r$ ) = *Lap-dist*  $e$   $r$   
 ⟨proof⟩

**lemma** *qbs-Lap-mechanism-qbs-l-inverse*: *Lap-mechanism* $_Q$   $e$   $x = \text{qbs-l-inverse}$  (*Lap-dist*  
 $e$   $x$ )  
 ⟨proof⟩

**proposition** *qbs-DP-Lap-mechanism*:

**assumes**  $\varepsilon > 0$  **and**  $|x - y| \leq r$

**shows** *DP-divergence* $_Q$  (*Lap-mechanism* $_Q$  ( $1 / \varepsilon$ )  $x$ ) (*Lap-mechanism* $_Q$  ( $1 / \varepsilon$ )  
 $y$ ) ( $r * \varepsilon$ ) = 0

⟨proof⟩

## 2.3 Naive Report Noisy Max Mechanism

**primrec** *qbs-NaiveRNM* :: *real*  $\Rightarrow$  *real list*  $\Rightarrow$  *real qbs-measure* **where**

*qbs-NaiveRNM*  $\varepsilon$  [] = *return-qbs borel* 0 |

*qbs-NaiveRNM*  $\varepsilon$  ( $x \# xs$ ) =

(*case*  $xs$  of

*Nil*  $\Rightarrow$  *Lap-mechanism* $_Q$  ( $1 / \varepsilon$ )  $x$  |

$y \# ys \Rightarrow \text{do } \{x1 \leftarrow \text{Lap-mechanism}_Q (1 / \varepsilon) x; x2 \leftarrow \text{qbs-NaiveRNM } \varepsilon xs;$   
*return-qbs borel* ( $\max x1 x2$ )}

**lemma** *qbs-morphism-NaiveRNM*[qbs]: *qbs-NaiveRNM*  $\in \text{borel}_Q \Rightarrow_Q \text{list-qbs borel}$   
 $\Rightarrow_Q \text{monadP-qbs borel}_Q$

*<proof>*

**theorem** *qbs-DP-NaiveRNM'*:

**assumes** *pos[arith,simp]*:  $\varepsilon > 0$

**and** *length xs = n and length ys = n*

**and** *adj*:  $(\sum i < n. |nth\ xs\ i - nth\ ys\ i|) \leq r$

**shows** *DP-divergence<sub>Q</sub> (qbs-NaiveRNM  $\varepsilon$  xs) (qbs-NaiveRNM  $\varepsilon$  ys) (r \*  $\varepsilon) = 0$*

*<proof>*

**definition** *adj-naive-RNM* :: *real  $\Rightarrow$  (real list  $\times$  real list) set* **where**

*adj-naive-RNM r  $\equiv$  {(xs,ys). length xs = length ys  $\wedge$  ( $\sum i < length\ xs.$  |nth xs i - nth ys i|)  $\leq$  r}*

**theorem** *qbs-DP-NaiveRNM*:

**assumes** *pos*:  $\varepsilon > 0$

**shows** *differential-privacy<sub>Q</sub> (qbs-NaiveRNM  $\varepsilon$ ) (adj-naive-RNM r) (r \*  $\varepsilon) 0$*

*<proof>*

**end**

## References

- [1] T. Sato and S. Katsumata. Divergences on monads for relational program logics. *Mathematical Structures in Computer Science*, 33(45):427–485, 2023.