## Decision problems on probabilistic programs over finite fields and all their extensions

Charlie Jacomme supervised by Hubert Comon & Steve Kremer November 19,2019

LSV, INRIA Nancy

# This is NOT an introduction to security

**Probabilistic programs over finite fields** Loosely speaking, a program:

- receives input values inside a finite fields,
- performs random sampling,
- performs operations, branchings, ..., (no loops)
- returns some values inside the finite field.

**Probabilistic programs over finite fields** Loosely speaking, a program:

- receives input values inside a finite fields,
- performs random sampling,
- performs operations, branchings, ..., (no loops)
- returns some values inside the finite field.

 $\hookrightarrow$  Verifications for such programs ?

• EQUIV: Are two programs equivalent ?

- EQUIV: Are two programs equivalent ?
- INDEP: Are two programs independent ?

- EQUIV: Are two programs equivalent ?
- INDEP: Are two programs independent ?
- MAJ: Can we bound the probability of an event inside a program ?

- EQUIV: Are two programs equivalent ?
- INDEP: Are two programs independent ?
- MAJ: Can we bound the probability of an event inside a program ?

Decidable ?

- EQUIV: Are two programs equivalent ?
- INDEP: Are two programs independent ?
- MAJ: Can we bound the probability of an event inside a program ?

#### Decidable ?

Given a finite field, everything is finite and can be computed.

- EQUIV: Are two programs equivalent ?
- INDEP: Are two programs independent ?
- MAJ: Can we bound the probability of an event inside a program ?

#### Decidable ?

Given a finite field, everything is finite and can be computed.

 $\hookrightarrow$  What about the complexity ?

### Uniform Verification of Probabilistic Programs

**Uniform Decisions problems** 

Given probabilistic programs over  $\mathbb{F}_2$  (booleans), using  $\oplus$  and  $\wedge,$  are they equivalent for all length of bitstrings ?

Given probabilistic programs over  $\mathbb{F}_2$  (booleans), using  $\oplus$  and  $\wedge,$  are they equivalent for all length of bitstrings ?

 $EQUIV_{2^{\infty}}$ 

input: two programs over  $\mathbb{F}_2$ 

question: for any  $k \geq 1$ , are the programs equivalent over  $\mathbb{F}_{2^k}$  ?

Given probabilistic programs over  $\mathbb{F}_2$  (booleans), using  $\oplus$  and  $\wedge,$  are they equivalent for all length of bitstrings ?

 $\mathsf{EQUIV}_{2^\infty}$ 

input: two programs over  $\mathbb{F}_2$ 

question: for any  $k\geq 1$ , are the programs equivalent over  $\mathbb{F}_{2^k}$  ?

 $\hookrightarrow$  the length of the bitstrings could be a security parameter

Given probabilistic programs over  $\mathbb{F}_2$  (booleans), using  $\oplus$  and  $\wedge,$  are they equivalent for all length of bitstrings ?

 $EQUIV_{2^{\infty}}$ 

input: two programs over  $\mathbb{F}_2$ 

question: for any  $k\geq 1$ , are the programs equivalent over  $\mathbb{F}_{2^k}$  ?

 $\hookrightarrow$  the length of the bitstrings could be a security parameter

Decidable ?

Given probabilistic programs over  $\mathbb{F}_2$  (booleans), using  $\oplus$  and  $\wedge,$  are they equivalent for all length of bitstrings ?

 $\mathsf{EQUIV}_{2^\infty}$ 

input: two programs over  $\mathbb{F}_2$ 

question: for any  $k \geq 1$ , are the programs equivalent over  $\mathbb{F}_{2^k}$  ?

 $\hookrightarrow$  the length of the bitstrings could be a security parameter

#### Decidable ?

There is an infinite number of cases to check, not so trivial anymore...

 $\mathsf{INDEP}_q \Leftrightarrow \mathsf{EQUIV}_q$ 

 $\begin{aligned} \mathsf{INDEP}_q \Leftrightarrow \mathsf{EQUIV}_q \\ \mathsf{NI}-\mathsf{EQUIV}_q \Leftrightarrow \mathsf{EQUIV}_q \end{aligned}$ 

 $\begin{aligned} \mathsf{INDEP}_q \Leftrightarrow \mathsf{EQUIV}_q \\ \mathsf{NI}-\mathsf{EQUIV}_q \Leftrightarrow \mathsf{EQUIV}_q \end{aligned}$ 

 $\begin{aligned} \mathsf{INDEP}_q \Leftrightarrow \mathsf{EQUIV}_q \\ \mathsf{NI}-\mathsf{EQUIV}_q \Leftrightarrow \mathsf{EQUIV}_q \end{aligned}$ 

	EQUIV <sub>x</sub>	$NI - MAJ_x$	$MAJ_x$
x = q	$coNP^{C_=P}$ -complete	PP-complete	$coNP^{PP}$ -complete
$x = q^{\infty}$	EXP coNP <sup>C_P</sup> -hard	$\leq_{EXP} POSITIVITY$	?

Complexity Menu<sup>1</sup> Formal definitions \*\*\* Definition of  $C_{-P}$ EQUIV<sub>*a*</sub> is  $coNP^{C_{=}P}$ -complete \*\*\* Decidability of EQUIV<sub> $a^{\infty}$ </sub> (without inputs/conditionals)

<sup>&</sup>lt;sup>1</sup>Familiarity with coNP complexity, completeness and oracles TM appreciated.

Complexity Menu<sup>1</sup> Formal definitions \* \* \*Definition of C=P EQUIV<sub>q</sub> is coNP<sup>C=P</sup>-complete \* \* \*Decidability of EQUIV<sub>q</sub><sup>∞</sup> (without inputs/conditionals) Decidability Menu Formal definitions  $\star \star \star$ Removing the inputs Decidability of EQUIV<sub>q</sub> $_{\infty}$ (without inputs/conditionals) Removing the conditionals

<sup>&</sup>lt;sup>1</sup>Familiarity with coNP complexity, completeness and oracles TM appreciated.

## Formal definitions

Classically, p denotes a prime number,  $q = p^k$  a prime power.

Classically, p denotes a prime number,  $q = p^k$  a prime power. There is a unique finite field for each size. Classically, p denotes a prime number,  $q = p^k$  a prime power. There is a unique finite field for each size.

Prime finite field

 $\mathbb{F}_p \simeq \mathbb{Z}/p\mathbb{Z}$  (integers modulo p)

Classically, p denotes a prime number,  $q = p^k$  a prime power. There is a unique finite field for each size.

Prime finite field

 $\mathbb{F}_p \simeq \mathbb{Z}/p\mathbb{Z}$  (integers modulo p)

#### **Finite fields**

With  $\alpha$  indeterminate and  $P \in \mathbb{F}_{p}[\alpha]$  an irreducible polynomial of degree k:

 $\mathbb{F}_q \simeq \mathbb{F}_p[\alpha] / P(\alpha)$ 

We consider in this talk:

 $\begin{array}{ll} e & ::= & S & & \text{a polynomial over } \mathbb{F}_q[I \uplus R] \\ & | & \text{if } S = 0 \text{ then } e_1 \text{ else } e_2 & \text{conditionals} \end{array}$ 

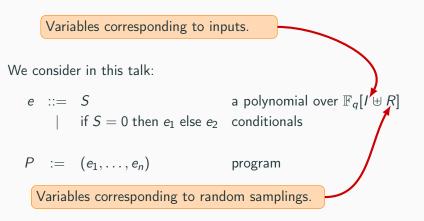
 $P := (e_1, \ldots, e_n)$  program

Variables corresponding to inputs.

We consider in this talk:

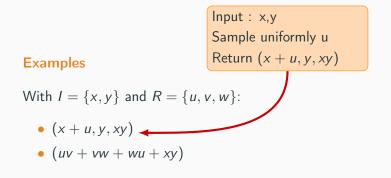
 $e ::= S \qquad \text{a polynomial over } \mathbb{F}_q[I \uplus R]$  $| \quad \text{if } S = 0 \text{ then } e_1 \text{ else } e_2 \quad \text{conditionals}$ 

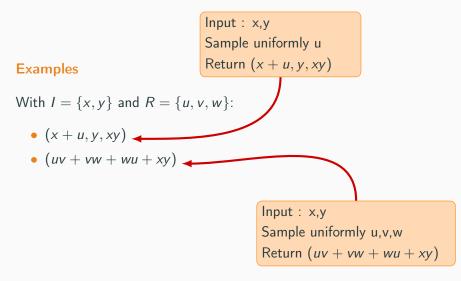
$$P := (e_1, \dots, e_n)$$
 program



#### **Examples**

With  $I = \{x, y\}$  and  $R = \{u, v, w\}$ :





We denote  $\mathcal{P}_q(I, R)$  the set of programs with input variables I and random variables R, and |P| the arity of P.

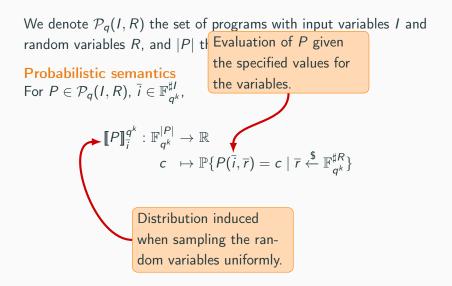
We denote  $\mathcal{P}_q(I, R)$  the set of programs with input variables I and random variables R, and |P| the arity of P.

Probabilistic semantics For  $P \in \mathcal{P}_q(I, R)$ ,  $\overline{i} \in \mathbb{F}_{q^k}^{\sharp I}$ ,

$$\llbracket P \rrbracket_{\overline{i}}^{q^{k}} : \mathbb{F}_{q^{k}}^{|P|} \to \mathbb{R}$$

$$c \quad \mapsto \mathbb{P}\{P(\overline{i}, \overline{r}) = c \mid \overline{r} \xleftarrow{\$} \mathbb{F}_{q^{k}}^{\sharp R}\}$$

We denote  $\mathcal{P}_q(I, R)$  the set of programs with input variables I and random variables R, and |P| the Evaluation of P given the specified values for the variables. For  $P \in \mathcal{P}_q(I, R)$ ,  $\overline{i} \in \mathbb{F}_{q^k}^{\sharp I}$ , the variables.  $\llbracket P \rrbracket_{\overline{i}}^{q^k} : \mathbb{F}_{q^k}^{|P|} \to \mathbb{R}$  $c \mapsto \mathbb{P}\{P(\overline{i}, \overline{r}) = c \mid \overline{r} \stackrel{\$}{\leftarrow} \mathbb{F}_{q^k}^{\sharp R}\}$ 



Over the booleans With  $I = \{i\}$  and  $R = \{r\}$ ,

$$\llbracket r \rrbracket^2 : 0 \mapsto \frac{1}{2} \\ 1 \mapsto \frac{1}{2}$$

Over the booleans With  $I = \{i\}$  and  $R = \{r\}$ ,  $\llbracket r \rrbracket^2 : 0 \mapsto \frac{1}{2}$   $1 \mapsto \frac{1}{2}$   $\llbracket ir \rrbracket^2_1 : 0 \mapsto \frac{1}{2}$   $\llbracket ir \rrbracket^2_0 : 0 \mapsto 1$   $1 \mapsto \frac{1}{2}$   $I \mapsto 0$ 

## Equivalence

$$egin{aligned} & P pprox_{q^k} & Q \ & \Leftrightarrow \ & orall ar{i} \in \mathbb{F}_{q^k}^{\sharp I}. \ \llbracket P 
brace_{ar{i}}^{q^k} = \llbracket Q 
brace_{ar{i}}^{q^k} \end{aligned}$$

#### Equivalence

$$egin{aligned} & P pprox_{q^k} & Q \ & \Leftrightarrow \ & orall ar{i} \in \mathbb{F}_{q^k}^{\sharp I}. \ \llbracket P 
brace_{ar{i}}^{q^k} = \llbracket Q 
brace_{ar{i}}^{q^k} \end{aligned}$$

**Uniform Equivalence** 

$$egin{array}{c} Ppprox_{q^{\infty}} & Q \ \Leftrightarrow \ orall k\geq 1. \ Ppprox_{q^k} & Q \end{array}$$

# The complexity of equivalence

#### SAT

input:  $\phi$  a CNF boolean formula question: Is  $\phi$  true for some valuation ?

#### SAT

input:  $\phi$  a CNF boolean formula

question: Is  $\phi$  true for some valuation ?

- L is in NP if
  - there exists a non deterministic TM, such that

 $x \in L \Leftrightarrow M(x)$  has at least one accepting path

#### SAT

input:  $\phi$  a CNF boolean formula question: Is  $\phi$  true for some valuation ?

- L is in NP if
  - there exists a non deterministic TM, such that

 $x \in L \Leftrightarrow M(x)$  has at least one accepting path

• there exists a probabilistic TM, such that

 $x \in L \Leftrightarrow M(x)$  accepts with non zero probability

#### halfSAT

input:  $\phi$  a CNF boolean formula

question: Is  $\phi$  true for for exactly half of its valuations ?

#### halfSAT

input:  $\phi$  a CNF boolean formula

question: Is  $\phi$  true for for exactly half of its valuations ?

- L is in  $C_{=}P$  if
  - there exists a non deterministic TM, such that

 $x \in L \Leftrightarrow$  half of the paths of M(x) are acceptings ones

#### halfSAT

input:  $\phi$  a CNF boolean formula

question: Is  $\phi$  true for for exactly half of its valuations ?

- L is in  $C_{=}P$  if
  - there exists a non deterministic TM, such that

 $x \in L \Leftrightarrow$  half of the paths of M(x) are acceptings ones

• there exists a probabilistic TM, such that

$$x \in L \Leftrightarrow M(x)$$
 accepts with probability  $\frac{1}{2}$ 

A-halfSAT

input:  $\phi(X, Y)$  a CNF boolean formula

question: For any valuation of Y,

is  $\phi$  true for for exactly half of the valuations of X ?

A-halfSAT

input:  $\phi(X, Y)$  a CNF boolean formula

question: For any valuation of Y,

is  $\phi$  true for for exactly half of the valuations of X ?

*L* is in  $coNP^{C_{=}P}$  if

A-halfSAT

input:  $\phi(X, Y)$  a CNF boolean formula

question: For any valuation of Y,

is  $\phi$  true for for exactly half of the valuations of X ?

- L is in  $coNP^{C_{=}P}$  if
  - There exists a non deterministic TM with an oracle deciding problems in C<sub>=</sub>P, such that

 $x \in L \Leftrightarrow$  all the paths of M(x) are acceptings ones

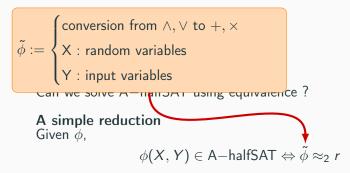
#### Can we solve A-halfSAT using equivalence ?

## Can we solve A-halfSAT using equivalence ?

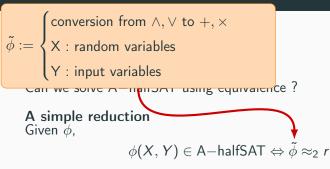
# A simple reduction Given $\phi$ ,

$$\phi(X,Y) \in \mathsf{A-halfSAT} \Leftrightarrow \tilde{\phi} \approx_2 r$$

### Hardness



## Hardness



 $\hookrightarrow \mathsf{EQUIV}_2$  is  $\mathsf{coNP}^{\mathsf{C}=\mathsf{P}}$ -hard

$$M(P, Q, c, \overline{i}) :=$$

$$x \stackrel{\$}{\leftarrow} \{0, 1\}$$

$$\overline{r} = (r_1, \dots, r_m) \stackrel{\$}{\leftarrow} \mathbb{F}_q^m$$
if  $x = 0$  then
if  $P(\overline{i}, \overline{r}) = c$  then ACCEPT else REJECT
else
if  $Q(\overline{i}, \overline{r}) = c$  then ACCEPT else REJECT

$$M(P, Q, c, \overline{i}) := \begin{cases} x \stackrel{\$}{\leftarrow} \{0, 1\} \\ \overline{r} = (r_1, \dots, r_m) \stackrel{\$}{\leftarrow} \mathbb{F}_q^m \\ \text{if } x = 0 \text{ then} \\ \text{if } P(\overline{i}, \overline{r}) = c \text{ then ACCEPT else REJECT} \\ \text{else} \\ \text{if } Q(\overline{i}, \overline{r}) = c \text{ then ACCEPT else REJECT} \end{cases}$$

$$\mathbb{P}_{accept}(P,Q,c,\overline{i}) = \frac{\llbracket P \rrbracket_{\overline{i}}^{q}(c) + \llbracket Q \rrbracket_{\overline{i}}^{q}(c)}{2}$$

$$M(P, Q, c, \overline{i}) :=$$

$$x \stackrel{\$}{\leftarrow} \{0, 1\}$$

$$\overline{r} = (r_1, \dots, r_m) \stackrel{\$}{\leftarrow} \mathbb{F}_q^m$$
if  $x = 0$  then  
if  $P(\overline{i}, \overline{r}) = c$  then ACCEPT else REJECT  
else  
if  $Q(\overline{i}, \overline{r}) = c$  then ACCEPT else REJECT  

$$\mathbb{P}_{accept}(P, Q, c, \overline{i}) = \frac{\llbracket P \rrbracket_{\overline{i}}^q(c) + \llbracket Q \rrbracket_{\overline{i}}^q(c)}{2}$$

Probablity that *P* equals *c* on input  $\overline{i}$ 

$$M(P, Q, c, \overline{i}) := \begin{vmatrix} x \stackrel{\$}{\leftarrow} \{0, 1\} \\ \overline{r} = (r_1, \dots, r_m) \stackrel{\$}{\leftarrow} \mathbb{F}_q^m \\ \text{if } x = 0 \text{ then} \\ \text{if } P(\overline{i}, \overline{r}) = c \text{ then ACCEPT else REJECT} \\ \text{else} \\ \text{if } Q(\overline{i}, \overline{r}) = c \text{ then ACCEPT else REJECT} \end{cases}$$

$$\mathbb{P}_{accept}(P,Q,c,\bar{i}) = \frac{\llbracket P \rrbracket_{\bar{i}}^{q}(c) + \llbracket Q \rrbracket_{\bar{i}}^{q}(c)}{2}$$

$$M(P, Q, c, \overline{i}) := \begin{cases} x \stackrel{\$}{\leftarrow} \{0, 1\} \\ \overline{r} = (r_1, \dots, r_m) \stackrel{\$}{\leftarrow} \mathbb{F}_q^m \\ \text{if } x = 0 \text{ then} \\ \text{if } P(\overline{i}, \overline{r}) = c \text{ then ACCEPT else REJECT} \\ \text{else} \\ \text{if } Q(\overline{i}, \overline{r}) = c \text{ then ACCEPT else REJECT} \end{cases}$$

$$\mathbb{P}_{accept}(P,Q,c,\bar{i}) = \frac{\llbracket P \rrbracket_{\bar{i}}^{q}(c) + \llbracket Q \rrbracket_{\bar{i}}^{q}(c)}{2}$$

$$M(P, Q, c, \overline{i}) :=$$

$$x \stackrel{\$}{\leftarrow} \{0, 1\}$$

$$\overline{r} = (r_1, \dots, r_m) \stackrel{\$}{\leftarrow} \mathbb{F}_q^m$$
if  $x = 0$  then  
if  $P(\overline{i}, \overline{r}) = c$  then ACCEPT else REJECT  
else  
if  $Q(\overline{i}, \overline{r}) \neq c$  then ACCEPT else REJECT

$$\mathbb{P}_{accept}(P,Q,c,\bar{i}) = \frac{\llbracket P \rrbracket_{\bar{i}}^{q}(c) + ?}{2}$$

$$M(P, Q, c, \overline{i}) := \begin{cases} x \stackrel{\$}{\leftarrow} \{0, 1\} \\ \overline{r} = (r_1, \dots, r_m) \stackrel{\$}{\leftarrow} \mathbb{F}_q^m \\ \text{if } x = 0 \text{ then} \\ \text{if } P(\overline{i}, \overline{r}) = c \text{ then ACCEPT else REJECT} \\ \text{else} \\ \text{if } Q(\overline{i}, \overline{r}) \neq c \text{ then ACCEPT else REJECT} \end{cases}$$

$$\mathbb{P}_{accept}(P, Q, c, \overline{i}) = \frac{\llbracket P \rrbracket_{\overline{i}}^{q}(c) + (1 - \llbracket Q \rrbracket_{\overline{i}}^{q}(c))}{2}$$

$$\mathbb{P}_{accept}(P, Q, c, \overline{i}) = \frac{1}{2} + \frac{\llbracket P \rrbracket_{\overline{i}}^{q}(c) - \llbracket Q \rrbracket_{\overline{i}}^{q}(c)}{2}$$

$$\mathbb{P}_{accept}(P,Q,c,\bar{i}) = \frac{1}{2} + \frac{\llbracket P \rrbracket_{\bar{i}}^{q}(c) - \llbracket Q \rrbracket_{\bar{i}}^{q}(c)}{2}$$

$$\mathbb{P}_{accept}(P,Q,c,\overline{i}) = \frac{1}{2} \Leftrightarrow \llbracket P \rrbracket_{\overline{i}}^{q}(c) = \llbracket Q \rrbracket_{\overline{i}}^{q}(c)$$

$$\mathbb{P}_{accept}(P,Q,c,\bar{i}) = \frac{1}{2} + \frac{\llbracket P \rrbracket_{\bar{i}}^q(c) - \llbracket Q \rrbracket_{\bar{i}}^q(c)}{2}$$

$$\mathbb{P}_{accept}(P,Q,c,ar{i}) = rac{1}{2} \Leftrightarrow \llbracket P 
rbracket_{ar{i}}^q(c) = \llbracket Q 
rbracket_{ar{i}}^q(c)$$

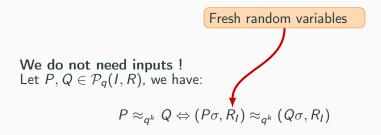
 $\hookrightarrow \text{ Given } P, Q, c, \overline{i}, \text{ deciding if } \llbracket P \rrbracket_{\overline{i}}^{q}(c) = \llbracket Q \rrbracket_{\overline{i}}^{q}(c) \text{ is in } C_{=} \mathsf{P}.$ 

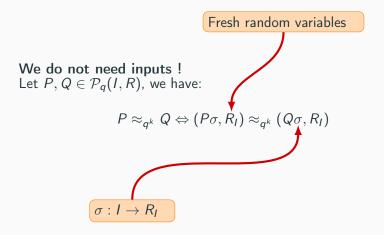
$$P \approx_q Q \Leftrightarrow \forall \overline{i} \in \mathbb{F}_q^m, \forall c \in \mathbb{F}_q^{|P|}, \llbracket P \rrbracket_{\overline{i}}^q(c) = \llbracket Q \rrbracket_{\overline{i}}^q(c)$$
$$\hookrightarrow \mathsf{EQUIV}_q \text{ is coNP}^{\mathsf{C}=\mathsf{P}}\text{-complete}$$

# Deciding uniform equivalence

We do not need inputs ! Let  $P, Q \in \mathcal{P}_q(I, R)$ , we have:

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$





We do not need inputs !

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$

 $P \approx_{q^k} Q$ 

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$

$$\begin{array}{l} P\approx_{q^k} Q\\ \Leftrightarrow \forall \overline{i}\in \mathbb{F}_{q^k}^{\sharp l}. \ \llbracket P \rrbracket_{\overline{i}}^{q^k} = \llbracket Q \rrbracket_{\overline{i}}^{q^k} \end{array}$$

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$

$$P \approx_{q^{k}} Q$$
  

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}} = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}$$
  

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \forall \overline{c} \in \mathbb{F}_{q^{k}}^{n}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}}(\overline{c}) = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}(\overline{c})$$

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$

$$P \approx_{q^{k}} Q$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}} = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \forall \overline{c} \in \mathbb{F}_{q^{k}}^{n}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}}(\overline{c}) = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}(\overline{c})$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \forall \overline{c} \in \mathbb{F}_{q^{k}}^{n}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}}(c, \overline{i}) = \llbracket (Q\sigma, R_{I}) \rrbracket_{\overline{i}}^{q^{k}}(c, \overline{i})$$

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$

$$P \approx_{q^{k}} Q$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}} = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \forall \overline{c} \in \mathbb{F}_{q^{k}}^{n}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}}(\overline{c}) = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}(\overline{c})$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \forall \overline{c} \in \mathbb{F}_{q^{k}}^{n} \llbracket (P\sigma, R_{I}) \rrbracket_{\overline{i}}^{q^{k}}(c, \overline{i}) = \llbracket (Q\sigma, R_{I}) \rrbracket_{\overline{i}}^{q^{k}}(c, \overline{i})$$

$$\Leftrightarrow \forall c' \in \mathbb{F}_{q^{k}}^{n+\sharp I} \llbracket (P\sigma, R_{I}) \rrbracket_{q^{k}}^{q^{k}}(c') = \llbracket (Q\sigma, R_{I}) \rrbracket_{q^{k}}^{q^{k}}(c')$$

$$P \approx_{q^k} Q \Leftrightarrow (P\sigma, R_I) \approx_{q^k} (Q\sigma, R_I)$$

$$P \approx_{q^{k}} Q$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}} = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \forall \overline{c} \in \mathbb{F}_{q^{k}}^{n}. \llbracket P \rrbracket_{\overline{i}}^{q^{k}}(\overline{c}) = \llbracket Q \rrbracket_{\overline{i}}^{q^{k}}(\overline{c})$$

$$\Leftrightarrow \forall \overline{i} \in \mathbb{F}_{q^{k}}^{\sharp I}. \forall \overline{c} \in \mathbb{F}_{q^{k}}^{n}. \llbracket (P\sigma, R_{I}) \rrbracket_{\overline{i}}^{q^{k}}(c, \overline{i}) = \llbracket (Q\sigma, R_{I}) \rrbracket_{\overline{i}}^{q^{k}}(c, \overline{i})$$

$$\Leftrightarrow \forall c' \in \mathbb{F}_{q^{k}}^{n+\sharp I} \llbracket (P\sigma, R_{I}) \rrbracket^{q^{k}}(c') = \llbracket (Q\sigma, R_{I}) \rrbracket^{q^{k}}(c')$$

$$\Leftrightarrow (P\sigma, R_{I}) \approx_{q^{k}} (Q\sigma, R_{I})$$

**Straight line programs** Programs without conditionals  $P, Q \in \mathcal{P}_q(\emptyset, R)$  **Straight line programs** Programs without conditionals  $P, Q \in \mathcal{P}_q(\emptyset, R)$ 

 $P, Q \in (\mathbb{F}_q[R])^n$ 

**Straight line programs** Programs without conditionals  $P, Q \in \mathcal{P}_q(\emptyset, R)$ 

$$P, Q \in (\mathbb{F}_q[R])^n$$

#### The mathematical question

$$P \approx_{q^{\infty}} Q$$

$$\Leftrightarrow$$

$$\forall \overline{c} \in \mathbb{F}_{q^{k}}^{n}. \ \forall k. \ \sharp\{\overline{r} \in \mathbb{F}_{q^{k}}^{\sharp R} | P(\overline{r}) = \overline{c}\} = \sharp\{\overline{r} \in \mathbb{F}_{q^{k}}^{\sharp R} | Q(\overline{r}) = \overline{c}\}$$

# The local zeta function For any $P \in \mathbb{F}_q[R]^n$ ,

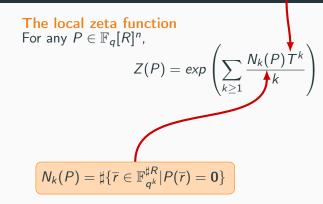
$$Z(P) = exp\left(\sum_{k\geq 1} \frac{N_k(P)T^k}{k}\right)$$

# An indeterminate

# The local zeta function For any $P \in \mathbb{F}_q[R]^n$ ,

$$Z(P) = exp\left(\sum_{k\geq 1} \frac{N_k(P)T^k}{k}\right)$$

#### An indeterminate



The local zeta function For any  $P \in \mathbb{F}_q[R]^n$ ,  $Z(P) = exp\left(\sum_{k \ge 1} \frac{N_k(P)T^k}{k}\right)$ 

Why is it interesting ?  $Z(P) = Z(Q) \Leftrightarrow \forall k. \ \sharp\{\overline{r} \in \mathbb{F}_{q^k}^{\sharp R} | P(\overline{r}) = \mathbf{0}\} = \sharp\{\overline{r} \in \mathbb{F}_{q^k}^{\sharp R} | Q(\overline{r}) = \mathbf{0}\}$ 

The local zeta function For any  $P \in \mathbb{F}_q[R]^n$ ,  $Z(P) = exp\left(\sum_{k \ge 1} \frac{N_k(P)T^k}{k}\right)$ 

Why is it interesting ?  $Z(P) = Z(Q) \Leftrightarrow \forall k. \ \sharp\{\bar{r} \in \mathbb{F}_{q^k}^{\sharp R} | P(\bar{r}) = \mathbf{0}\} = \sharp\{\bar{r} \in \mathbb{F}_{q^k}^{\sharp R} | Q(\bar{r}) = \mathbf{0}\}$ 

Black Magic By Well's conjecture (proven by Dwork), Z(P) is a rational function, and can thus be computed !

• using the local zeta function, we can decide if *P* and *Q* are equal to zero with the same probability on all extensions of a finite field;

- using the local zeta function, we can decide if *P* and *Q* are equal to zero with the same probability on all extensions of a finite field;
- with a bit of encoding, we can extend this to all values, and check the equality of the distributions.

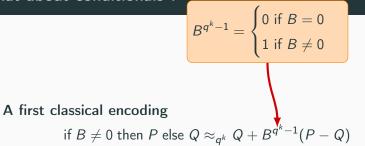
- using the local zeta function, we can decide if *P* and *Q* are equal to zero with the same probability on all extensions of a finite field;
- with a bit of encoding, we can extend this to all values, and check the equality of the distributions.

 $\hookrightarrow$  We can decide if  $P \approx_{q^{\infty}} Q$  !

#### A first classical encoding

 $\text{if }B\neq 0 \text{ then }P \text{ else }Q\approx_{q^k}Q+B^{q^k-1}(P-Q)$ 

# What about conditionals ?



# A first classical encoding

 $\text{if }B\neq 0 \text{ then }P \text{ else }Q\approx_{q^k}Q+B^{q^k-1}(P-Q)$ 

It depends on k...

- *B* has an inverse if and only if  $B \neq 0$ ,
- $\exists t, Bt 1 = 0$  if and only if  $B \neq 0$ ,
- for any variable *t* and polynomial *B*:

$$(B(Bt-1)=0 \wedge t(Bt-1)=0) \Leftrightarrow t=B^{q^k-2}$$

- *B* has an inverse if and only if  $B \neq 0$ ,
- $\exists t, Bt 1 = 0$  if and only if  $B \neq 0$ ,
- for any variable *t* and polynomial *B*:

$$(B(Bt-1)=0 \wedge t(Bt-1)=0) \Leftrightarrow t=B^{q^k-2}$$

 $\sharp\{\overline{r}\in\mathbb{F}_{q^k}^m,t\in\mathbb{F}_{q^k}\mid (Q+tB(P-Q),B(Bt-1),t(Bt-1))=\mathbf{0}\}$ 

- *B* has an inverse if and only if  $B \neq 0$ ,
- $\exists t, Bt 1 = 0$  if and only if  $B \neq 0$ ,
- for any variable t and polynomial B:

$$(B(Bt-1)=0 \wedge t(Bt-1)=0) \Leftrightarrow t=B^{q^k-2}$$

$$\begin{split} & \sharp\{\overline{r} \in \mathbb{F}_{q^k}^m, t \in \mathbb{F}_{q^k} \mid (Q + tB(P - Q), B(Bt - 1), t(Bt - 1)) = \mathbf{0}\} \\ & = \sharp\{\overline{r} \in \mathbb{F}_{q^k}^m \mid Q + B^{q^k - 1}(P - Q) = \mathbf{0}\} \end{split}$$

- *B* has an inverse if and only if  $B \neq 0$ ,
- $\exists t, Bt 1 = 0$  if and only if  $B \neq 0$ ,
- for any variable *t* and polynomial *B*:

$$(B(Bt-1)=0 \wedge t(Bt-1)=0) \Leftrightarrow t=B^{q^k-2}$$

$$\begin{split} & \sharp\{\overline{r} \in \mathbb{F}_{q^{k}}^{m}, t \in \mathbb{F}_{q^{k}} \mid (Q + tB(P - Q), B(Bt - 1), t(Bt - 1)) = \mathbf{0}\} \\ & = \sharp\{\overline{r} \in \mathbb{F}_{q^{k}}^{m} \mid Q + B^{q^{k} - 1}(P - Q) = \mathbf{0}\} \end{split}$$

 $\hookrightarrow$  We can use this inside  $N_k$ 

# Conclusions

# Uniform independence and equivalence

Decidable !

# Conclusions

#### Uniform independence and equivalence

Decidable !

Independence and equivalence

 $\mathsf{coNP}^{\mathsf{C}_{=}\mathsf{P}}\text{-}\mathsf{complete}...$ 

#### Uniform independence and equivalence

Decidable !

Independence and equivalence

 $\mathsf{coNP}^{\mathsf{C}_{=}\mathsf{P}}\text{-}\mathsf{complete}...$ 

#### Bounding the probability of an event

 $coNP^{PP}$ -complete in the finite case, reduces to the POSITIVITY problem in the uniform case.

### Uniform independence and equivalence

Decidable !

Independence and equivalence

 $\mathsf{coNP}^{\mathsf{C}_{=}\mathsf{P}}\text{-}\mathsf{complete}...$ 

**Bounding the probability of an event** coNP<sup>PP</sup>-complete in the finite case, reduces to the POSITIVITY problem in the uniform case.

> Given a sequence defined by a recurrence equations, are all the term of the sequence positive ?

# Uniform independence and equivalence

Decidable !

#### Independence and equivalence

 $\mathsf{coNP}^{\mathsf{C}_{=}\mathsf{P}}\text{-}\mathsf{complete}...$ 

**Bounding the probability of an event** coNP<sup>PP</sup>-complete in the finite case, reduces to the POSITIVITY problem in the uniform case.

# Which Programs ?

- Support of the observe primitive.
- sample variables inside a spolynomials,
- conditionals.

Given a sequence defined by a recurrence equations, are all the term of the sequence positive ?

#### The big question

Is the uniform problem strictly harder than the non uniform one ?

# The big question

Is the uniform problem strictly harder than the non uniform one ?

# Other open questions

- Can we support loops ?
- Is POSITIVITY decidable ?
- Can we extend to other probabilistic properties ?