On a composition theorem for randomized query complexity

Troy Lee
University of Technology Sydney

Joint work with: Dmitry Gavinsky, Miklos Santha, Swagato Sanyal

Function Composition

Say we have Boolean functions

$$g : S \to \{0,1\},\ \ S \subseteq \{0,1\}^m$$

$$f : \{0,1\}^n \to \{0,1\}$$

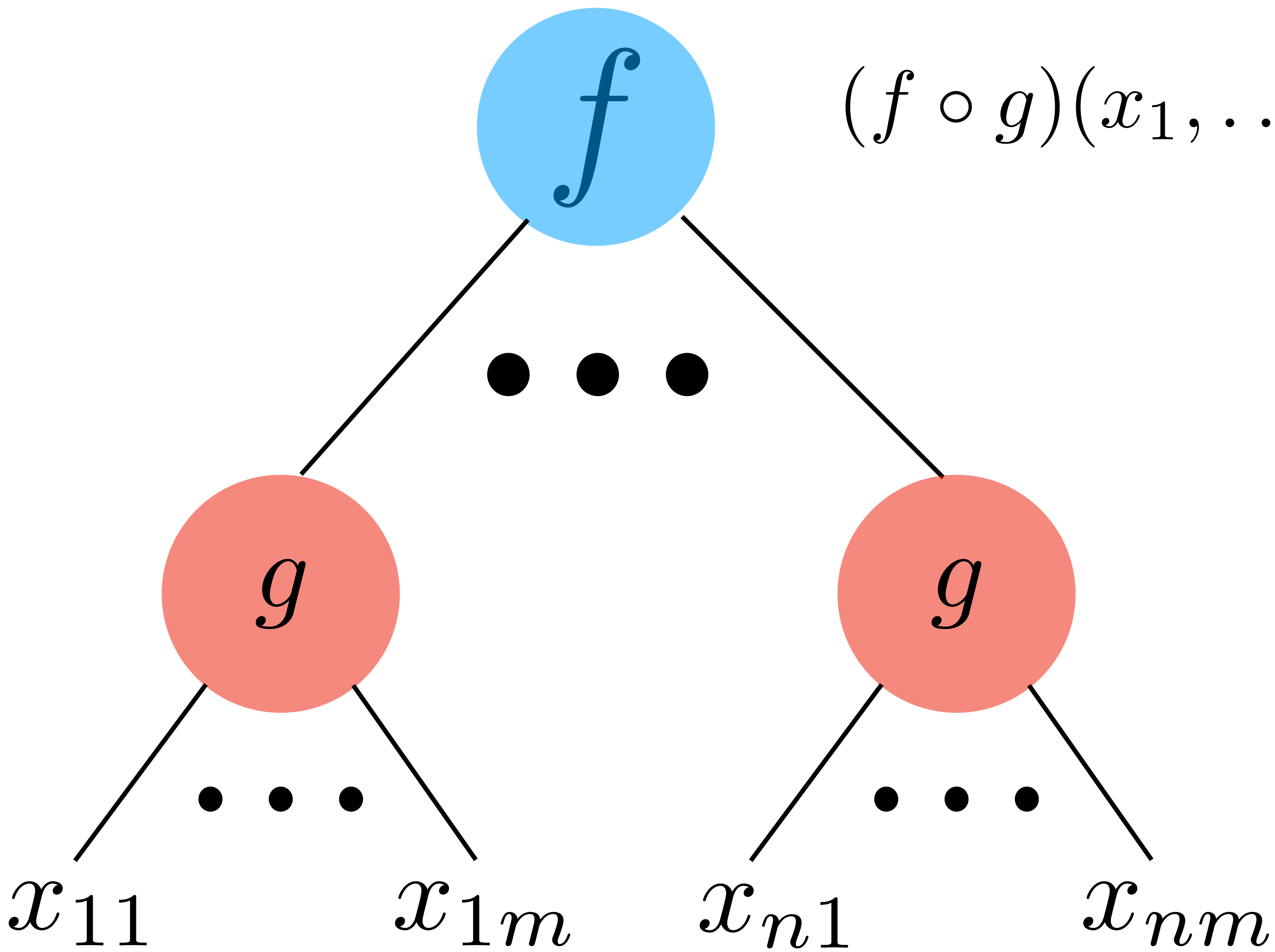
We can form the composition of these functions

$$f \circ g : S^n \to \{0,1\}$$

where

$$(f \circ g)(x_1, \ldots, x_n) = f(g(x_1), \ldots, g(x_n))$$

$$f \circ g : S^n \to \{0,1\}$$



$$(f \circ g)(x_1, \ldots, x_n) = f(g(x_1), \ldots, g(x_n))$$

What is the complexity of a composed function in terms of the complexities of f and g?

Typically a complexity measure $m(\cdot)$ is submultiplicative

$$m(f \circ g) \leq m(f) \cdot m(g)$$

The other (difficult) direction is a composition theorem.

- $\deg_{1/3}(AND_n \circ OR_m) = \Omega(\deg_{1/3}(AND_n) \cdot \deg_{1/3}(OR_m))$

  [Sherstov, Bun-Thaler]

- Lifting theorems: For the index communication gadget g

$$D^{cc}(f \circ g) = \Omega(D(f) \cdot D^{cc}(g))$$

$$R^{cc}_{1/3}(f \circ g) = \Omega(R_{1/3}(f) \cdot R^{cc}(g))$$

  [Goos, Pitassi, Watson]

- Composition behavior of certificate complexity, block sensitivity, etc. [Gilmer, Saks, Srinivasan and Tal]

Query Complexity

In this talk we focus on query complexity.

It is easy to see that for deterministic query complexity

$$D(f \circ g) \leq D(f) \cdot D(g)$$

Montanaro and (indep.) Tal show this is tight:

$$D(f \circ g) \geq D(f) \cdot D(g)$$

Deterministic query complexity perfectly composes.

Quantum Query Complexity

For quantum query complexity we also have a perfect composition theorem [Reichardt, Hoyer-L-Spalek]:

$$Q_{1/3}(f \circ g) = \Theta(Q_{1/3}(f) \cdot Q_{1/3}(g))$$

Randomized Query Complexity

The randomized case still remains open!

The easy direction holds:

$$R_{1/3}(f \circ g) = O(R_{1/3}(f) \cdot R_{1/3}(g) \cdot \log R_{1/3}(f))$$

What about a lower bound?

Randomized Composition Theorem

Ben-David and Kothari show the following lower bound:

$$R_{1/3}(f \circ g) = \Omega\left(R_{1/3}(f) \cdot \sqrt{\frac{R_{1/3}(g)}{\log R_{1/3}(g)}}\right)$$

for any partial function f and total function g.

The square root is strange!

We show an example where f is a relation and g is a partial function where the square root is needed.

Result

There is a relation f and partial function g such that

$$R_{1/3}(f \circ g) = O\left(R_{4/9}(f)\sqrt{R_{1/3}(g)}\right)$$

For any relation f and partial function g it holds that

$$R_{1/3}(f \circ g) = \Omega\left(R_{4/9}(f)\sqrt{R_{1/3}(g)}\right)$$

Example

First we describe the example.

The relation $f \subseteq \{0,1\}^n \times \{0,1\}^n$ is defined as

$$f = \{(x,a) : d_H(x,a) \leq n/2 - \sqrt{n}\}$$

Make queries to x and the goal is to output an a that agrees with x in at least $n/2 + \sqrt{n}$ positions.

Claim: $R_{4/9}(f) = \Theta(\sqrt{n})$.

Example

Take the inner partial function as $g : S \to \{0,1\}$,

$$S = \{x \in \{0,1\}^n : |x| \leq n/2 - \sqrt{n}\} \cup \{x \in \{0,1\}^n : |x| \geq n/2 + \sqrt{n}\}$$

$$g(x) = \begin{cases} 0 & \text{if } |x| \leq n/2 - \sqrt{n} \\ 1 & \text{if } |x| \geq n/2 - \sqrt{n} \end{cases}$$

Claim: $R_{1/3}(g) = \Theta(n)$.

Example

$$g(x) = \begin{cases} 0 & \text{if } |x| \leq n/2 - \sqrt{n} \\ 1 & \text{if } |x| \geq n/2 - \sqrt{n} \end{cases}$$

Claim: $R_{1/3}(g) = \Theta(n)$.

Proof: The gap hamming distance communication problem is

$$\mathrm{GHD}(x, y) = g(x \oplus y)$$

Chakrabarti and Regev '10 show $R^{cc}_{1/3}(\mathrm{GHD}) = \Omega(n)$, which implies the query complexity lower bound.

Observation

$$g(x) = \begin{cases} 0 & \text{if } |x| \leq n/2 - \sqrt{n} \\ 1 & \text{if } |x| \geq n/2 - \sqrt{n} \end{cases}$$

Claim: $R_{1/2-10/\sqrt{n}}(g) = O(1)$.

Proof: For any x in the domain of g,

$$\Pr_i[x_i = g(x)] = \frac{1}{2} + \frac{1}{\sqrt{n}}$$

Sample 100 bits randomly from x and take the majority vote.

$$f = \{(x,a) : d_H(x,a) \leq n/2 - \sqrt{n}\}$$

$$g(x) = \begin{cases} 0 & \text{if } |x| \leq n/2 - \sqrt{n} \\ 1 & \text{if } |x| \geq n/2 - \sqrt{n} \end{cases}$$

Here is a protocol for $f \circ g$ on input $x_1, \ldots, x_n$.

We want to output something close to $(g(x_1), \ldots, g(x_n))$.

For each $i = 1, \ldots, n$ let a_i be the majority of 100 randomly chosen bits from x_i. Output $a = a_1, \ldots, a_n$.

Each $a_i = g(x_i)$ with probability $1/2 + 10/\sqrt{n}$, so a agrees with x in at least $n/2 + \sqrt{n}$ positions with high probability by a Chernoff bound.

$$f = \{(x,a) : d_H(x,a) \le n/2 - \sqrt{n}\}$$

$$g(x) = \begin{cases} 0 & \text{if } |x| \le n/2 - \sqrt{n} \\ 1 & \text{if } |x| \ge n/2 - \sqrt{n} \end{cases}$$

We have given a protocol for $f \circ g$ with $O(n)$ queries.

Recall that $R_{4/9}(f) = \Omega(\sqrt{n}), R_{1/3}(g) = \Omega(n)$.

For this problem the bound

$$R_{1/3}(f \circ g) = \Omega\left(R_{4/9}(f)\sqrt{R_{1/3}(g)}\right)$$

is tight.

Lower Bound

$$R_{4/9}(f) = O\left(\frac{R_{1/3}(f \circ g)}{m(g)}\right)$$

Natural idea: use a protocol π for $f \circ g$ to give a protocol for f (also used by Ben-David and Kothari).

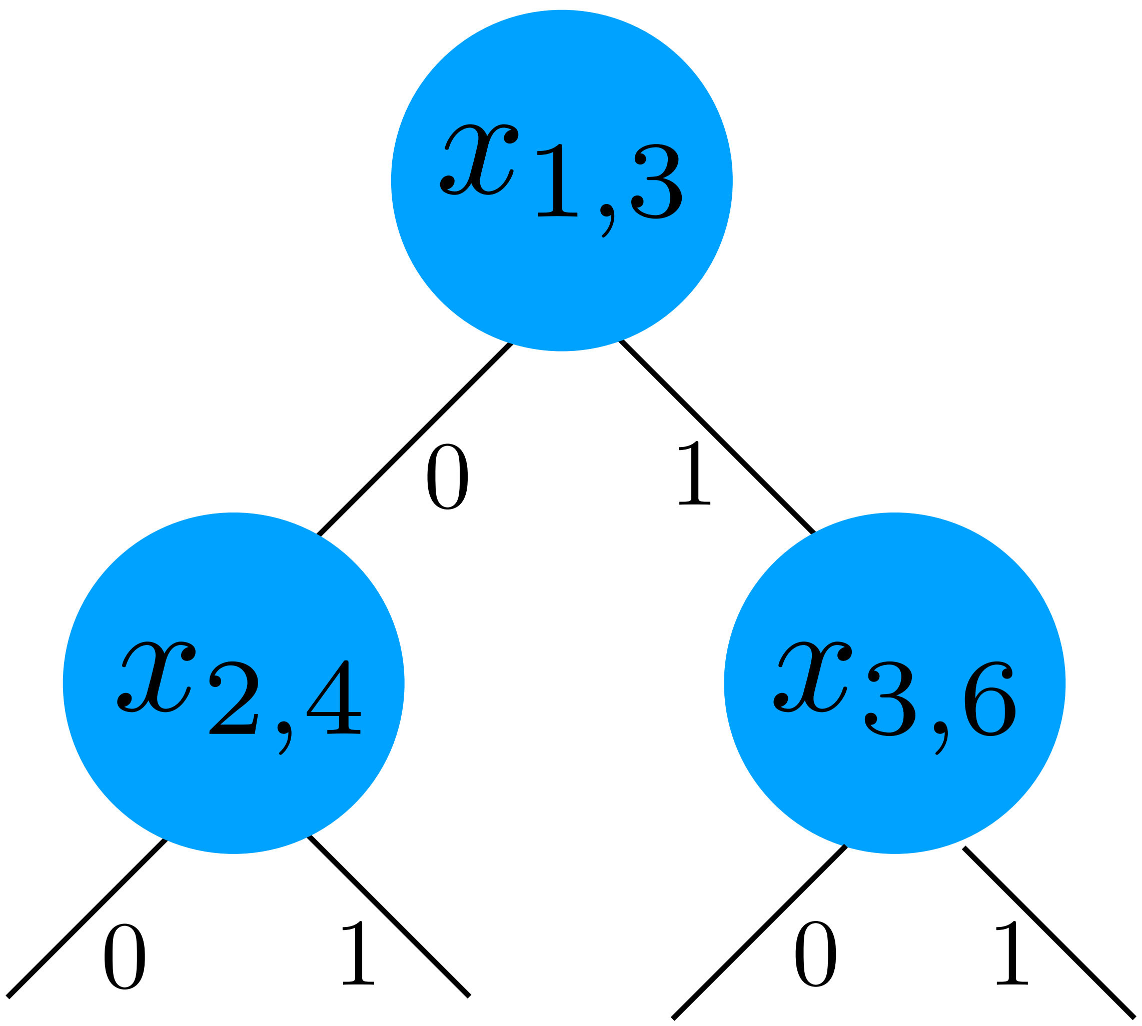
On input $z \in \{0,1\}^n$ sample $(x_1, \ldots, x_n)$ with $g(x_i) = z_i$ and run π on $(x_1, \ldots, x_n)$.

For any such $(x_1, \ldots, x_n)$, whp π outputs an a with $(z, a) \in f$.

Problem: Sampling x_i with $g(x_i) = z_i$ requires knowledge of z_i .

Let μ_0, μ_1 be distributions over $g^{-1}(0), g^{-1}(1)$, resp.

Fix $z \in \{0,1\}^n$ and sample $x_i \sim \mu_{z_i}$ for $i = 1, \ldots, n$.

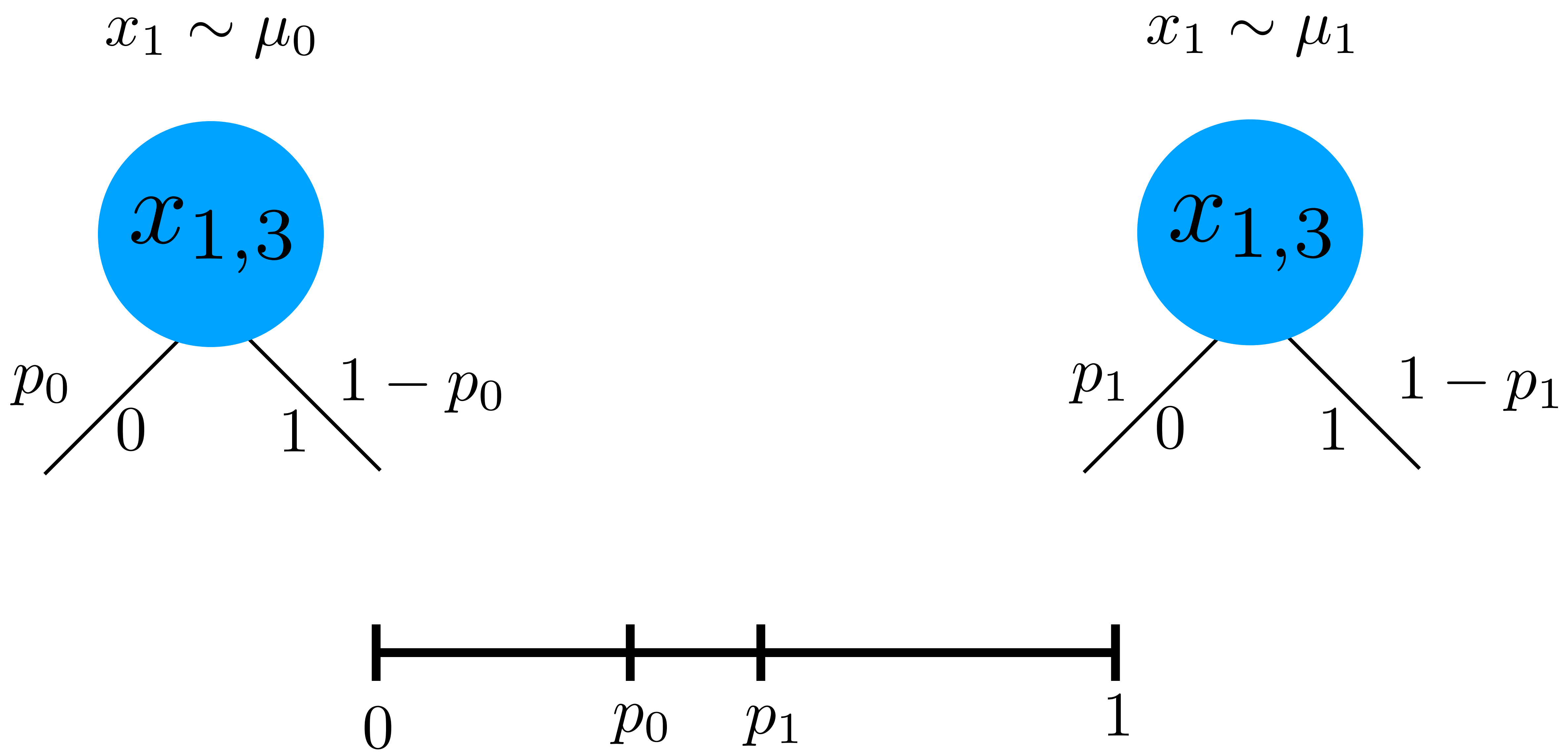


This induces a probability distribution over paths in the tree.

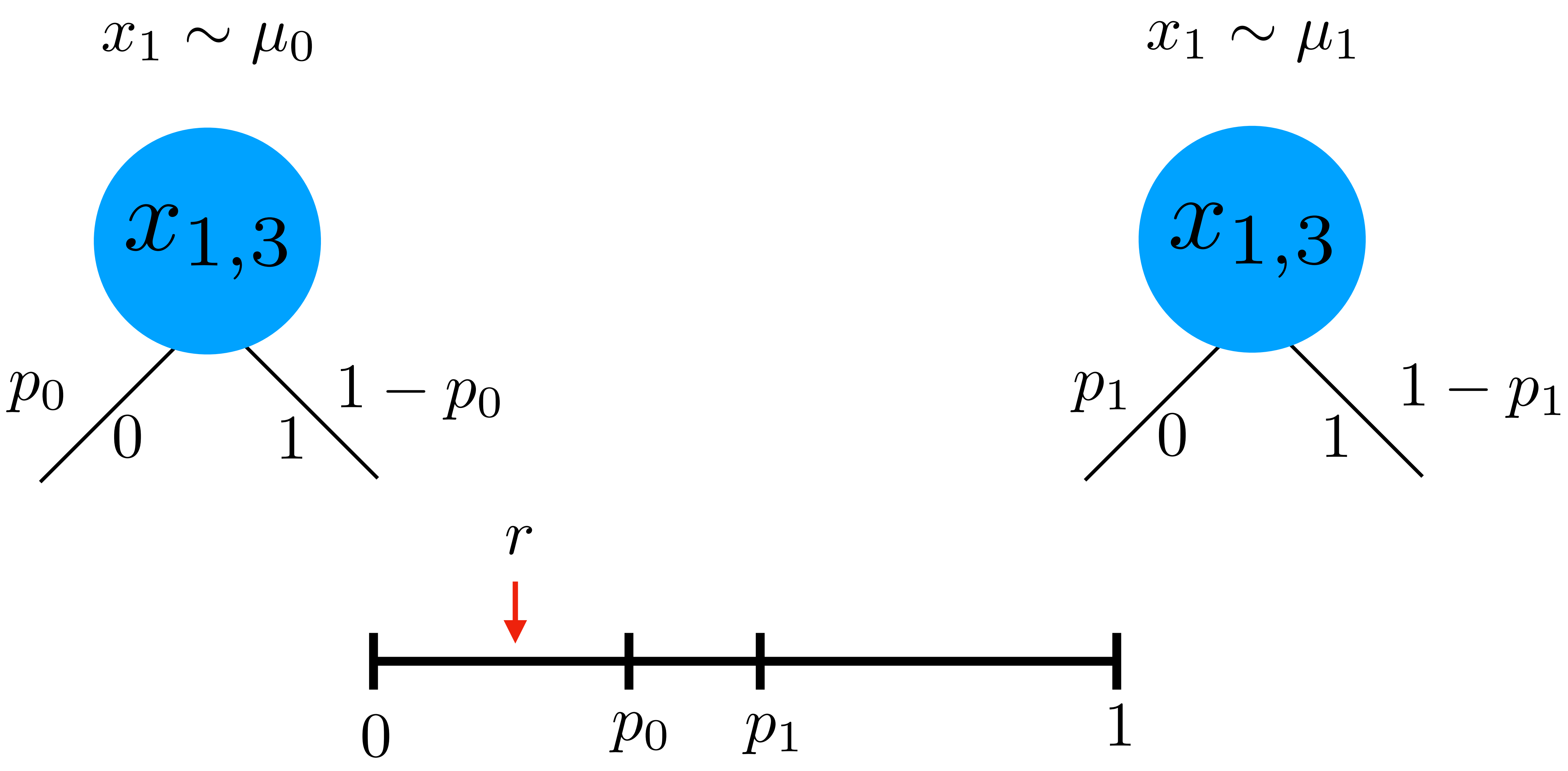
We want to simulate this distribution while querying as little as possible.

Deterministic tree for $f \circ g$.

Think about the first query in the tree for $f \circ g$.

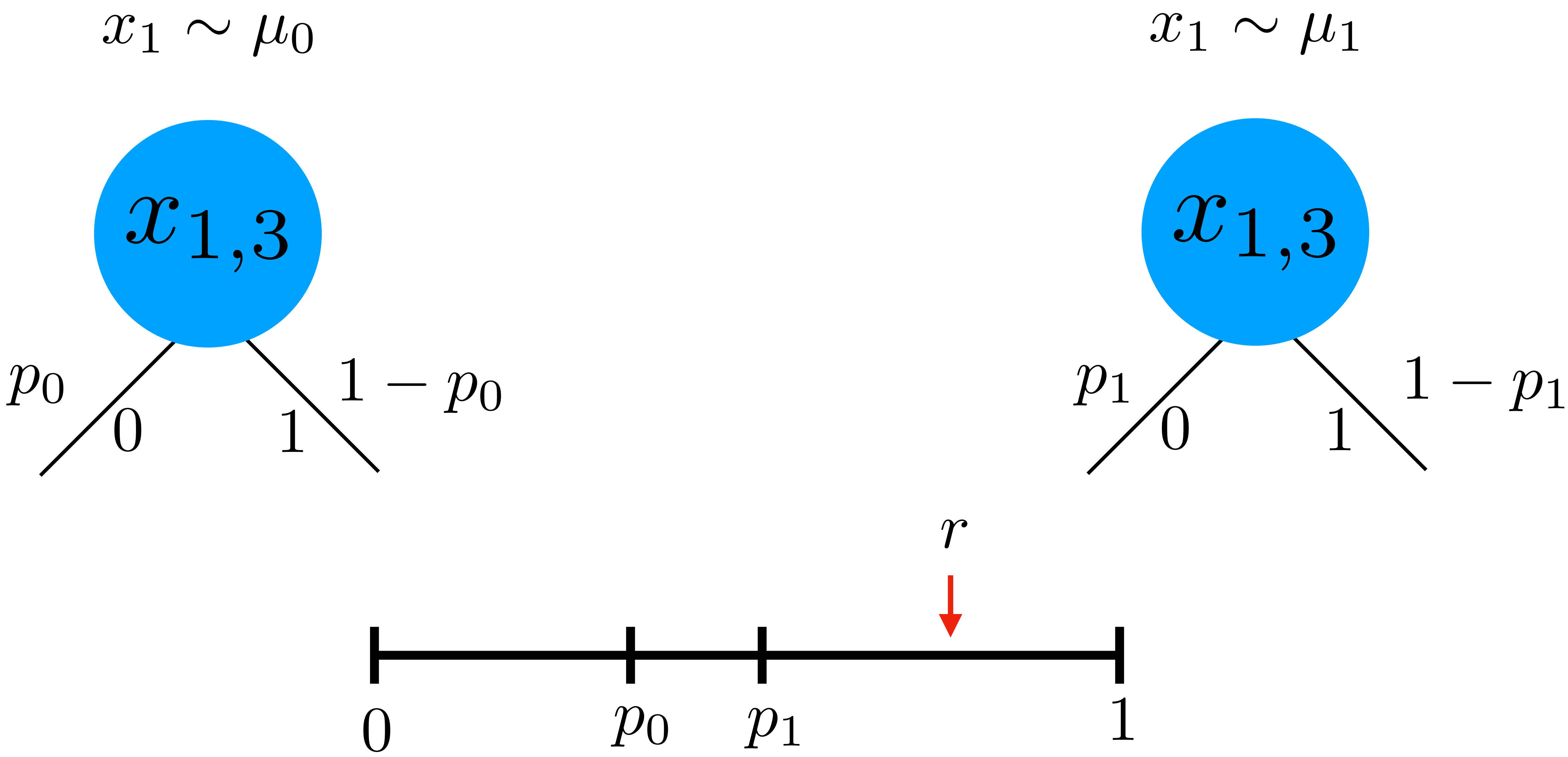


To simulate this query, uniformly choose $r \in [0, 1]$.



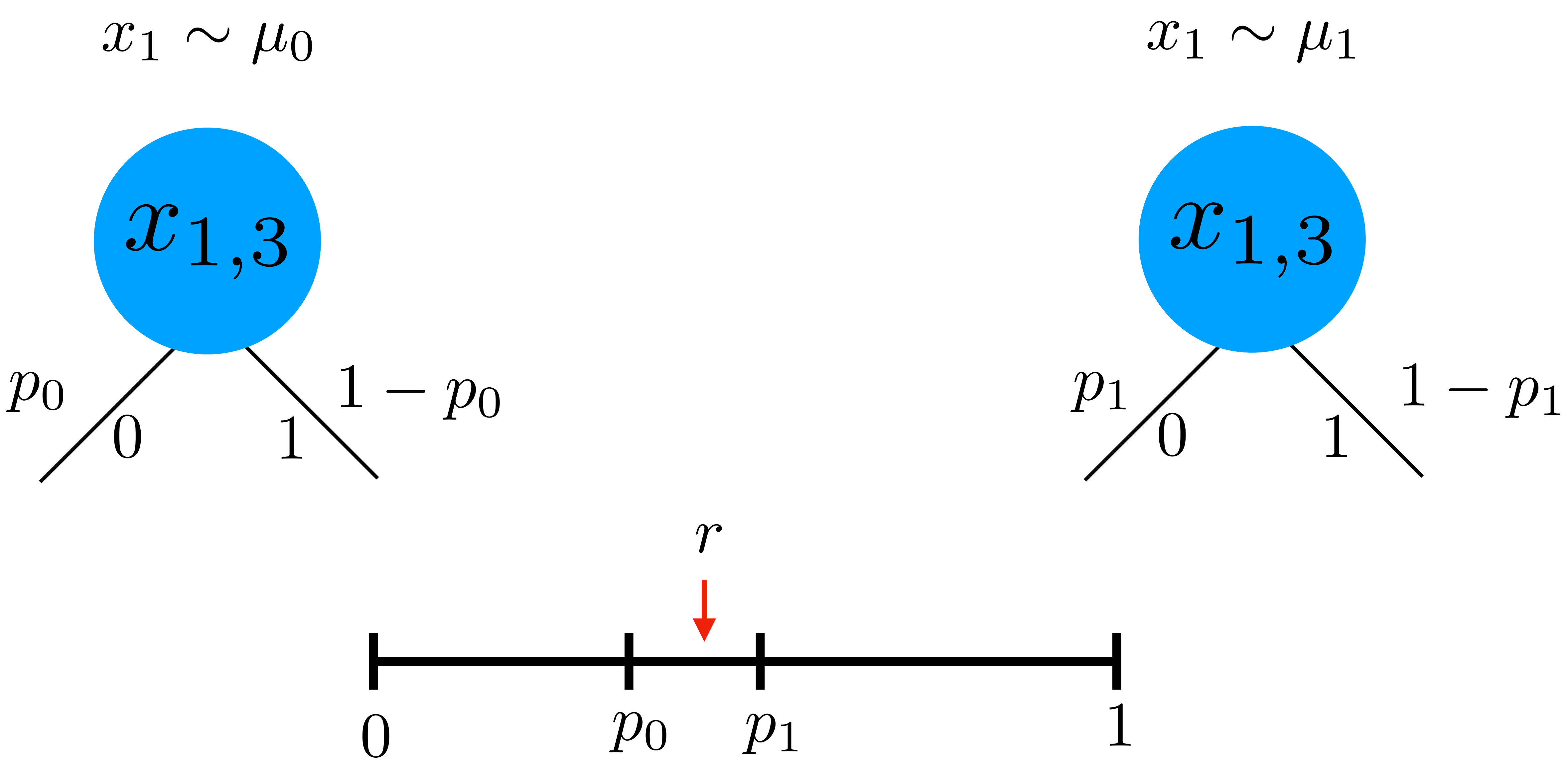
Bitsampler: uniformly choose $r \in [0, 1]$.

If $r \leq p_0$ answer $x_{1,3} = 0$.



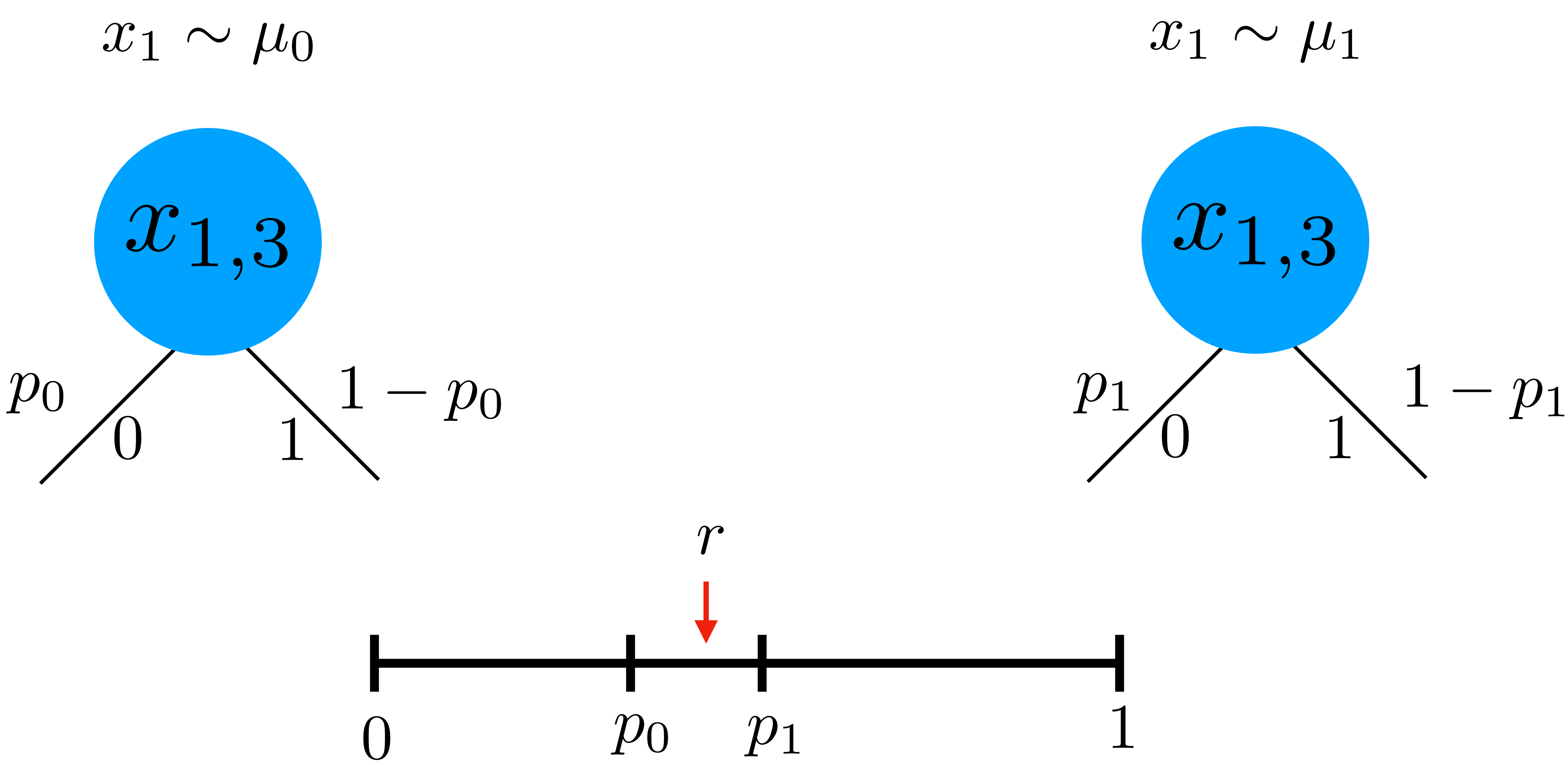
Bitsampler: uniformly choose $r \in [0, 1]$.

If $r > p_1$ answer $x_{1,3} = 1$.



Bitsampler: uniformly choose $r \in [0, 1]$.

If $p_0 < r \leq p_1$ then query z_1 and answer accordingly.



Bitsampler: uniformly choose $r \in [0, 1]$.

With this procedure we always move left with the correct probability.

Conflict Complexity

To analyze how many queries this algorithm makes we introduce the conflict complexity $\chi(g)$.

For a tree computing g and distributions μ_0, μ_1 look at expected number of times Bitsampler is run before making a query.

Maximize over distributions, minimize over trees = $\chi(g)$.

Wrapping up

With a direct sum theorem for conflict complexity we show

$$R_{4/9}(f) = O\left(\frac{R_{1/3}(f \circ g)}{\chi(g)}\right)$$

Conflict complexity is quadratically tight, even for partial g

$$\chi(g) = \Omega\left(\sqrt{R_{1/3}(g)}\right)$$

There exists an unbounded separation between sabotage complexity and $R_{1/3}(g)$ for a partial g.

Open Questions

What about the case where f, g are total functions?

How does conflict complexity compare to other lower bounds?

Is $\chi(g) = \min_{\epsilon} \frac{R_{1/2-\epsilon}(g)}{\epsilon}$?

[suggested by reviewer]