Probabilistic Program Analysis

Matthew B. Dwyer, Antonio Filieri, Jaco Geldenhuys, Mitchell Gerrard,
Corina Păsăreanu, and Willem Visser

Department of Computer Science and Engineering
University of Nebraska - Lincoln
Lincoln, Nebraska USA

August 2015

University of Nebraska Lincoln

Your view of "Program Analysis"

When you hear the term program analysis what comes to mind?

Your view of "Program Analysis"

When you hear the term program analysis what comes to mind?

What applications?

What kinds of information are computed?

What's the relationship between that information and program semantics?

How are the analysis results computed?

University of Nebraska Lincoln

Program analysis to me

I worked at a compiler company in the 1980s; responsible for the common optimizer in a family of C compilers for embedded systems

My Ph.D. work applied data flow analysis to check safety properties of concurrent programs

Co-directed a large project on software model checking (e.g., Bandera,Bogor)

Developed symbolic execution for program equivalence checking

Optimization of runtime monitoring for complex properties (e.g., state machines)

Verification of high-performance computing codes (e.g., MPI, OpenMP, CUDA, pthreads)

University of Nebraska Lincoln

What is a probabilistic program?

What is a probabilistic program?

Classic view is due to Kozen's 1978 semantics

- enrich your favorite language with the ability to draw from a probability distribution, e.g., `normal(0,1)`

What is a probabilistic program?

Classic view is due to Kozen's 1978 semantics

- enrich your favorite language with the ability to draw from a probability distribution, e.g., `normal(0,1)`

Pretty much every program is probabilistic in this sense; inputs are distributed in some way

What is a probabilistic program?

Classic view is due to Kozen's 1978 semantics

- enrich your favorite language with the ability to draw from a probability distribution, e.g., `normal(0,1)`

Pretty much every program is probabilistic in this sense; inputs are distributed in some way

Modern view is that probabilistic programs encode *probabilistic graphical models* which capture conditional dependence between random variables

- further enrich language with the ability to condition the output, e.g., `observe(x>0)`

What is a probabilistic program?

Classic view is due to Kozen's 1978 semantics

- enrich your favorite language with the ability to draw from a probability distribution, e.g., `normal(0,1)`

Pretty much every program is probabilistic in this sense; inputs are distributed in some way

Modern view is that probabilistic programs encode *probabilistic graphical models* which capture conditional dependence between random variables

- further enrich language with the ability to condition the output, e.g., `observe(x>0)`

The modern view is being driven by applications of machine learning

- Infering an input distribution from (enough) observed outputs amounts to a backward data flow analysis (Claret et al, FSE'13)

What is a probabilistic program?

Classic view is due to Kozen's 1978 semantics

- Enrich your favorite language with the ability to draw from a probability distribution, e.g., `normal(0,1)`

Pretty much every program is probabilistic in this sense; inputs are distributed in some way

Modern view is that probabilistic programs encode *probabilistic graphical models* which capture conditional dependence between random variables

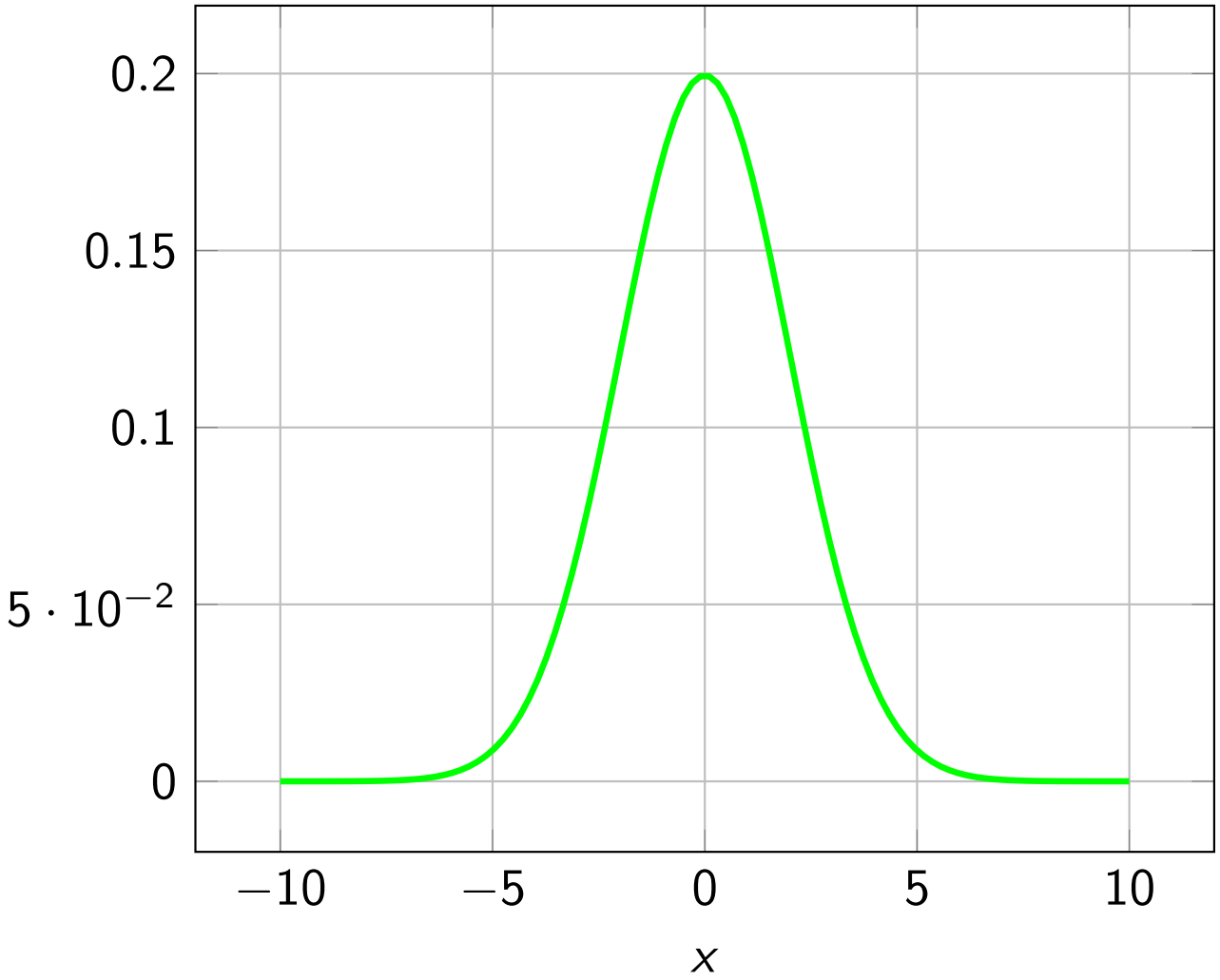
- Further enrich language with the ability to condition the output, e.g., `observe(x>0)`

The modern view is being driven by applications of machine learning

- Infering an input distribution from (enough) observed outputs amounts to a backward data flow analysis

University of Nebraska Lincoln

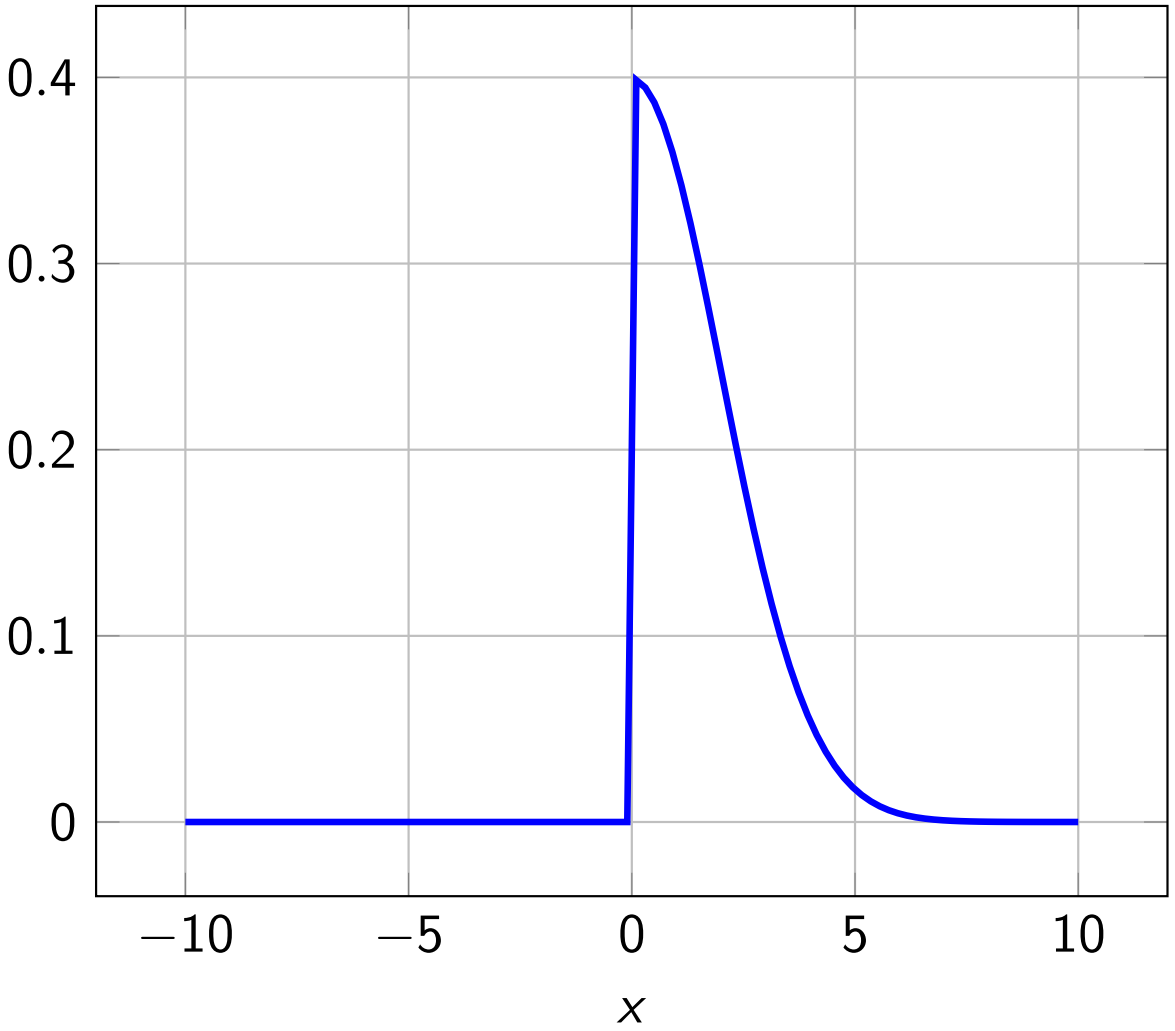
Imagine an input distributed according to $N(0, 2)$



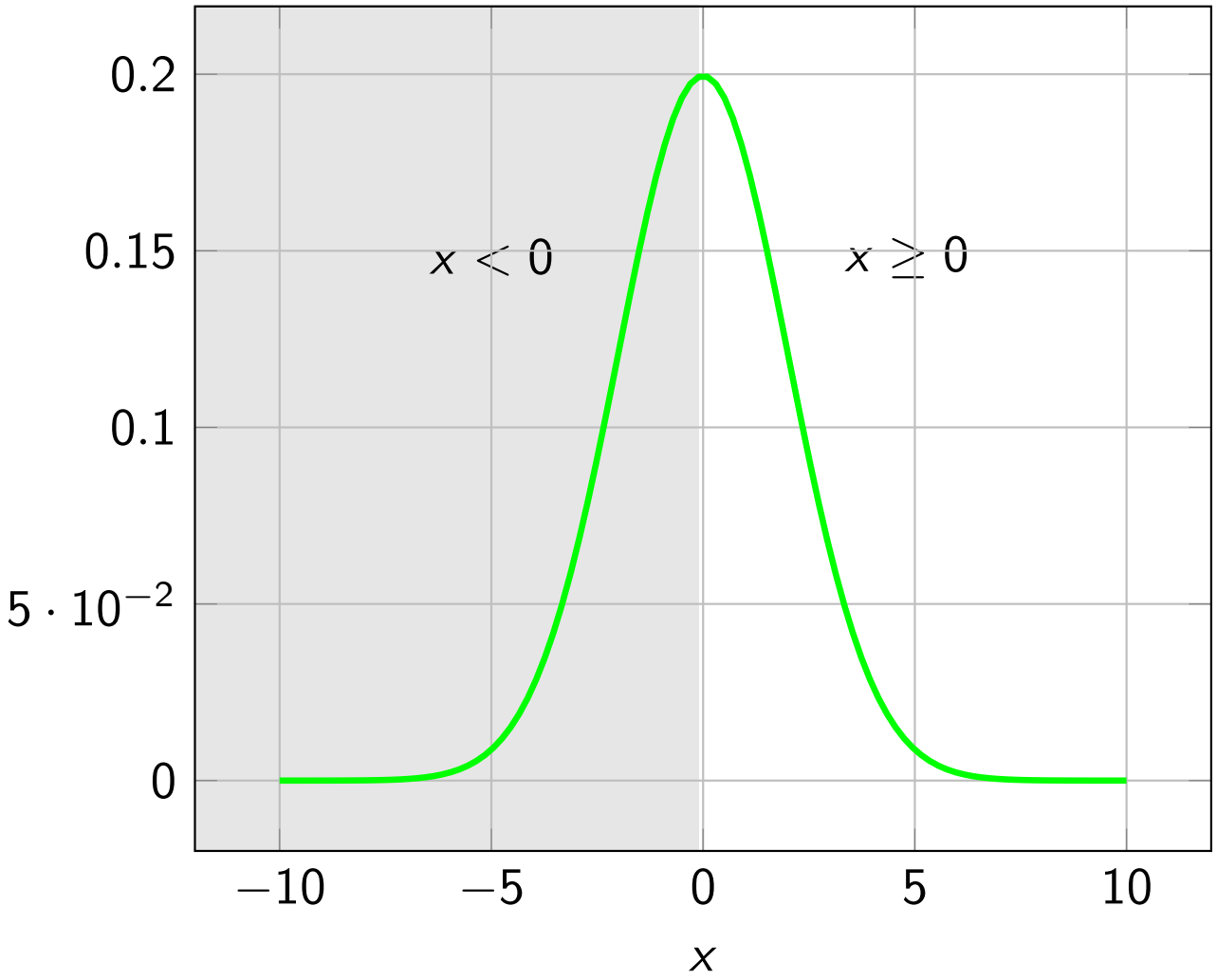
A trivial program

```
double abs(double x) {
  if (x<0)
    return -x;
  else
    return x;
}
```

Here is the output distribution

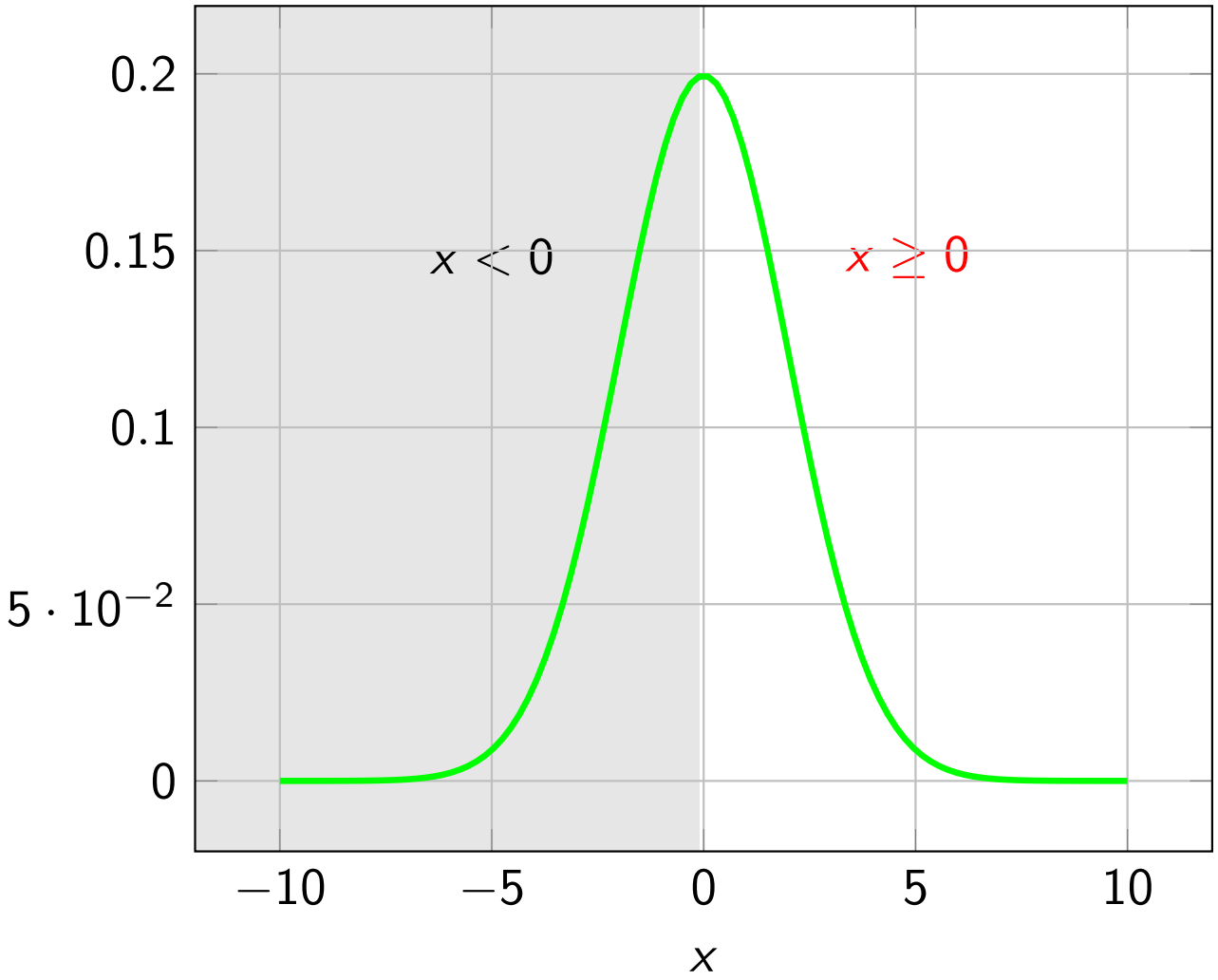


What's going on here?



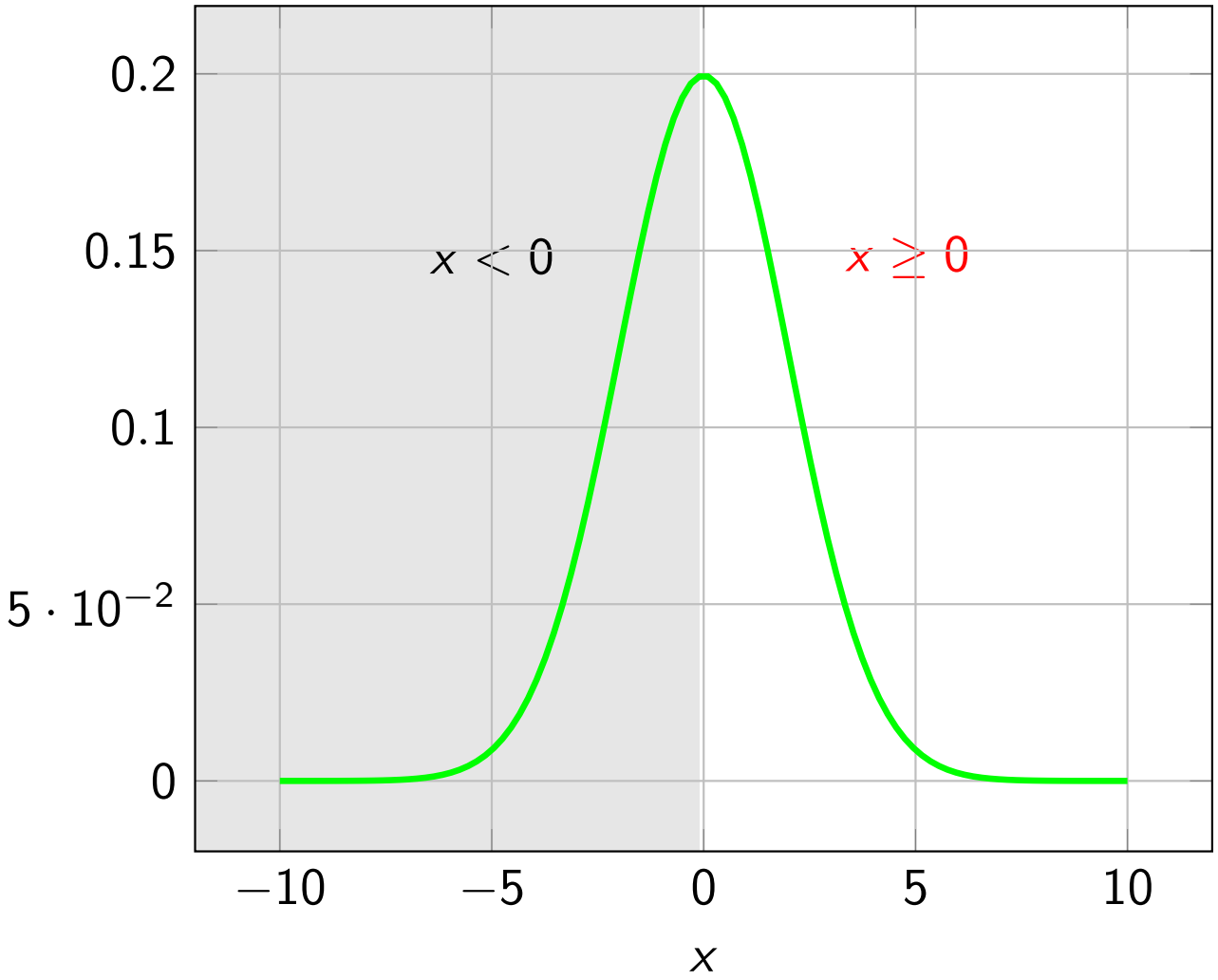
Let's think in terms of a very coarse division of the input

What's going on here?



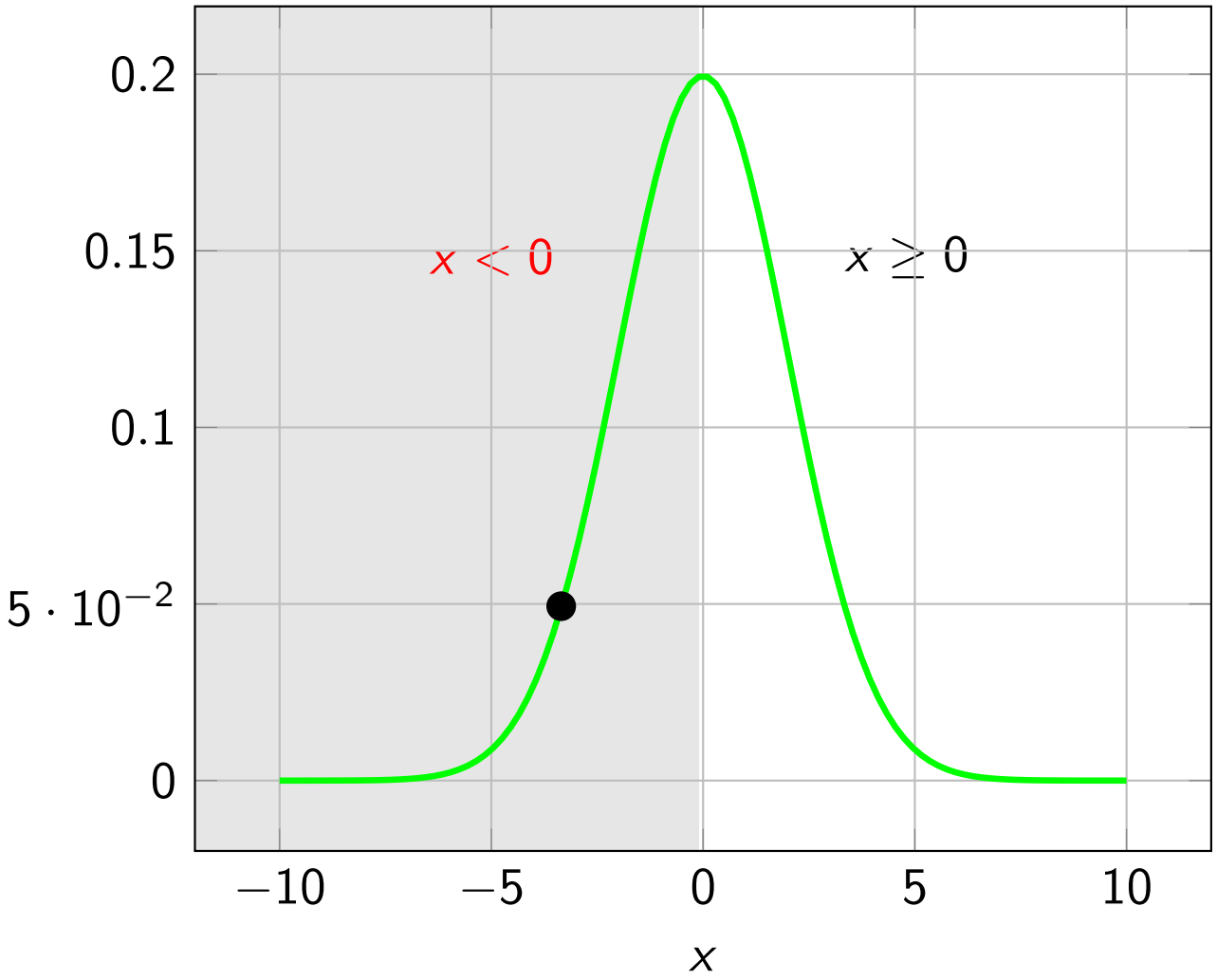
Input values $x \geq 0$ appear on the output unchanged.

What's going on here?



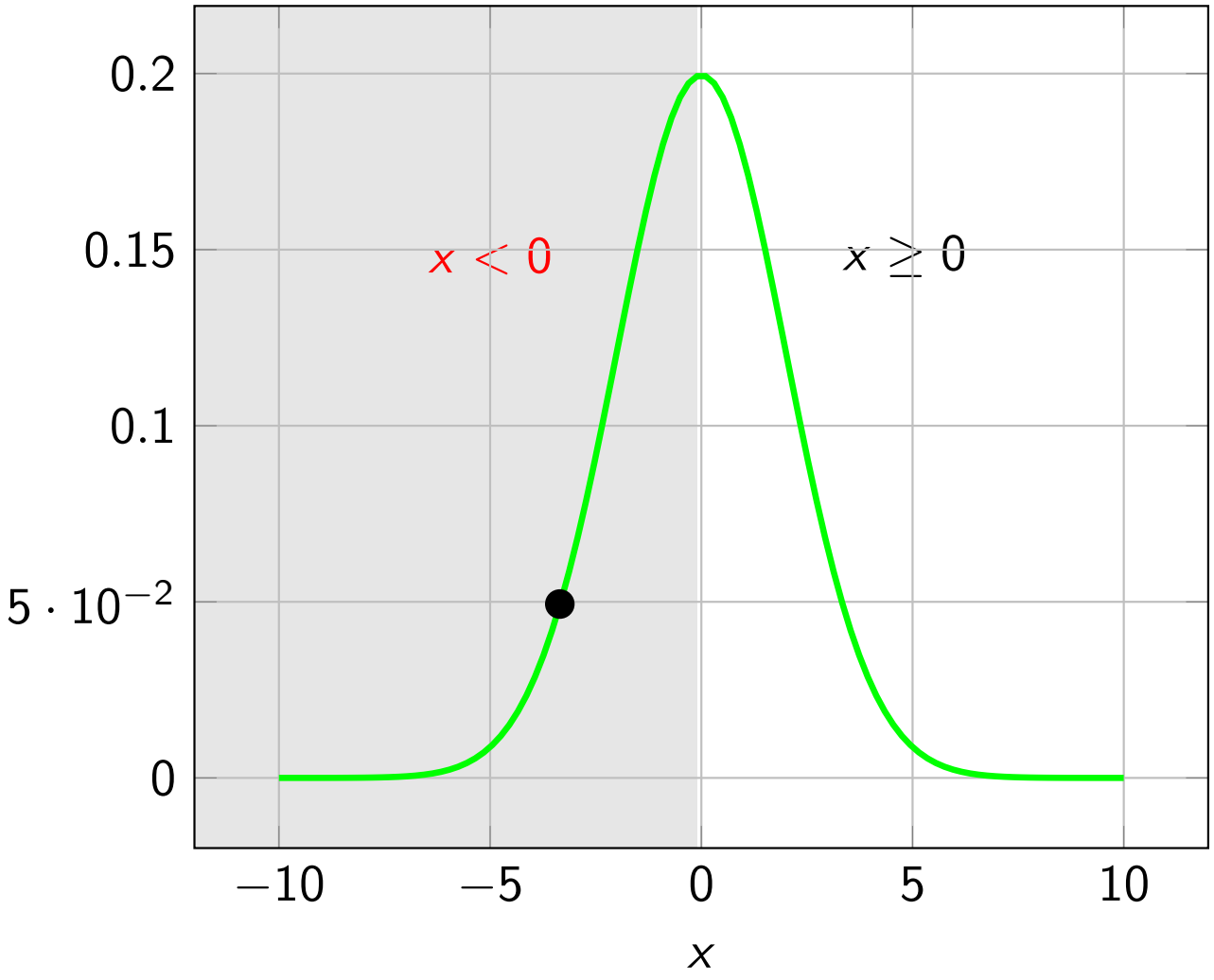
Input values $x \geq 0$ appear on the output unchanged.
Their mass in the input distribution propagates to the output.

What's going on here?



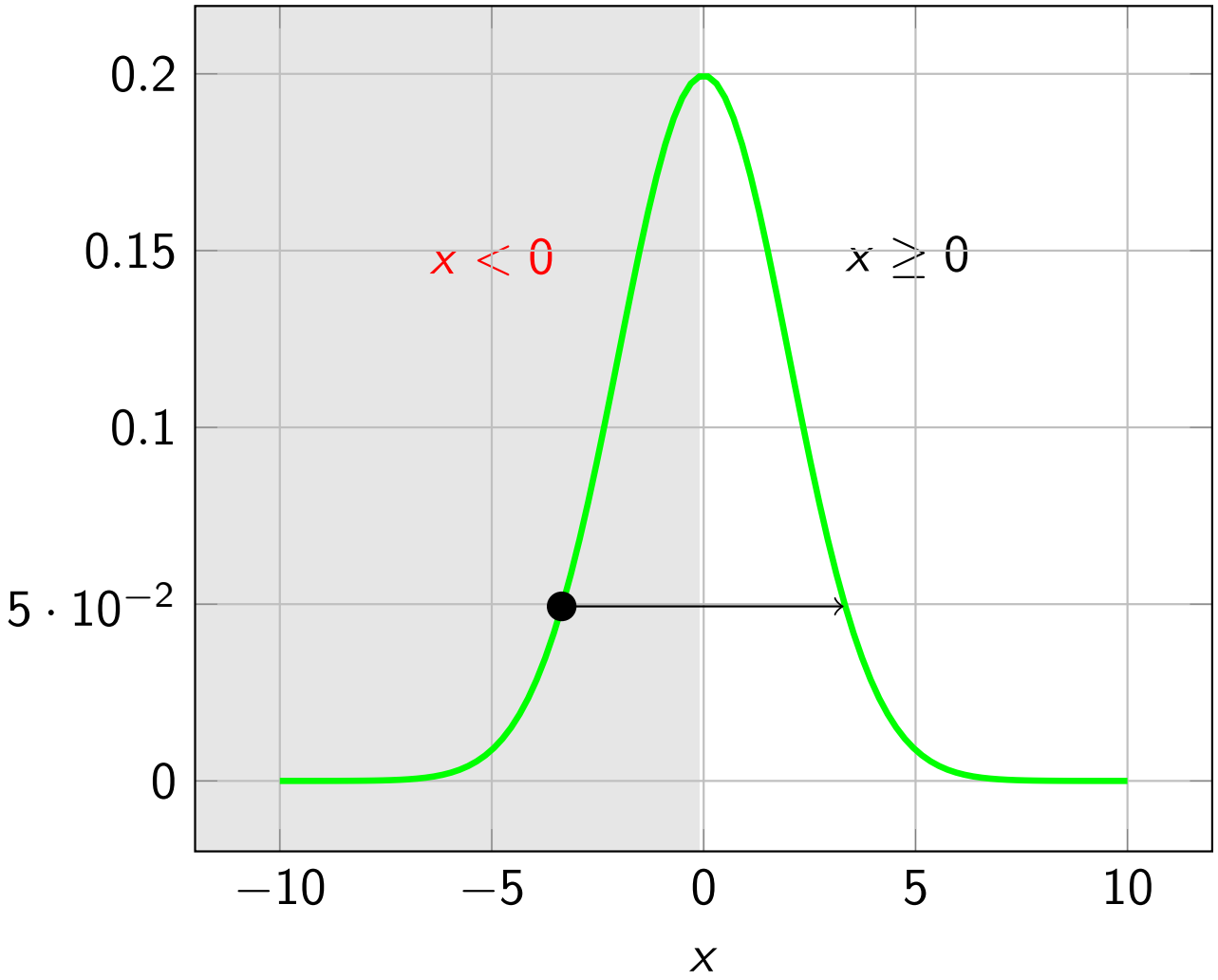
Input values $x < 0$ are transformed.

What's going on here?



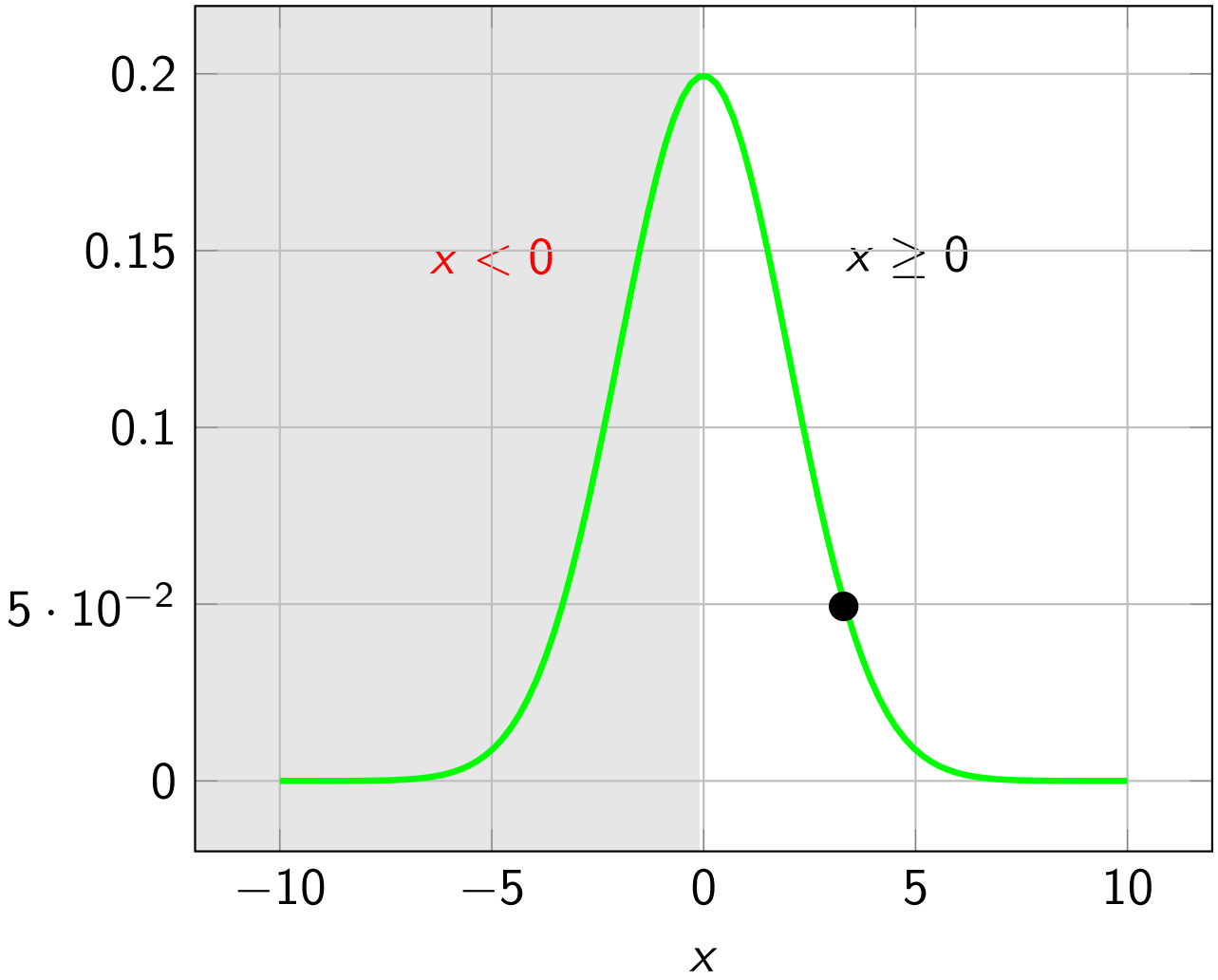
Input values $x < 0$ are transformed.
Their mass in the input distribution is shifted to $-x$

What's going on here?



Input values $x < 0$ are transformed.
Their mass in the input distribution is shifted to $-x$

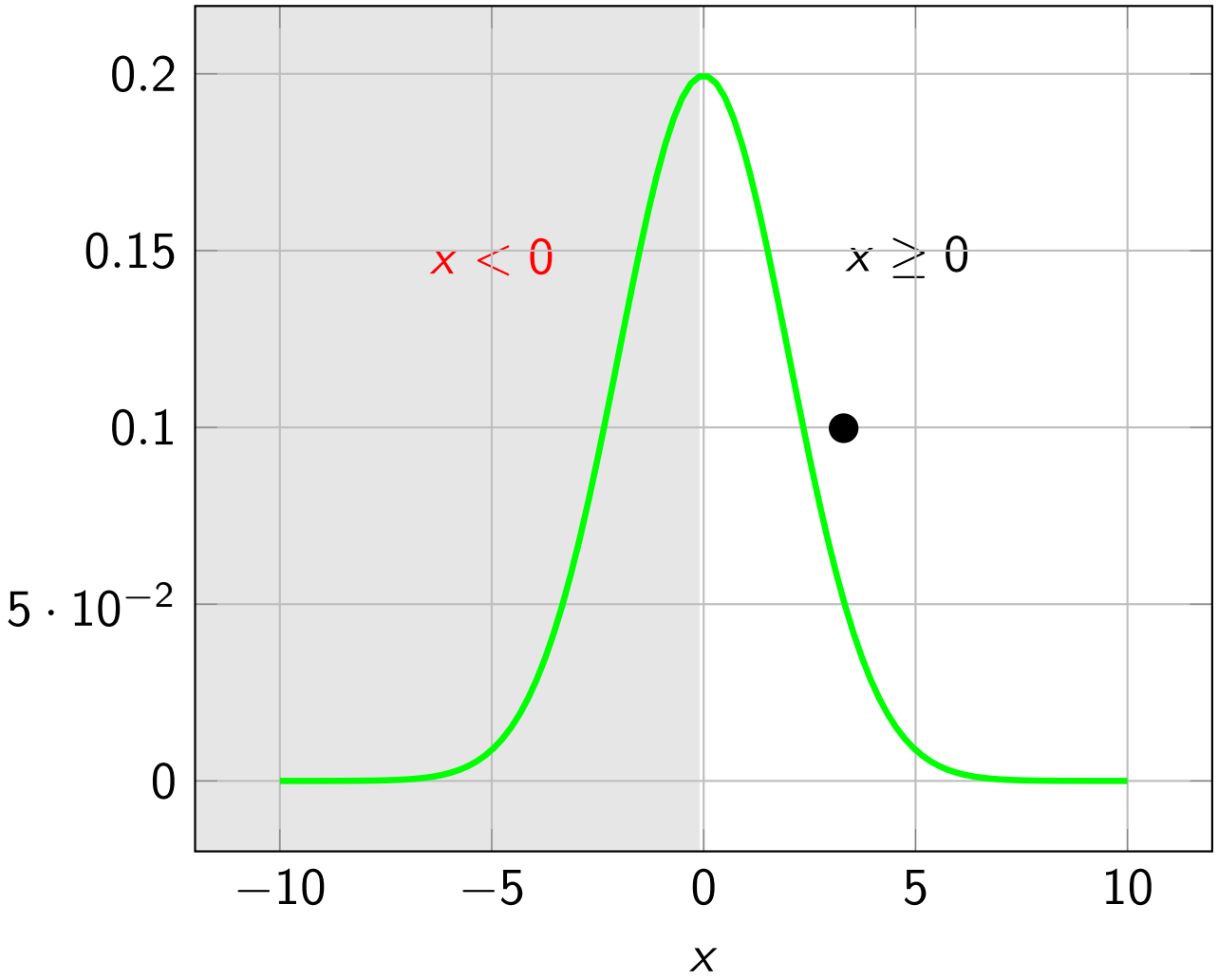
What's going on here?



Input values $x < 0$ are transformed.
Their mass in the input distribution is shifted to $-x$ and accumulates in the output distribution

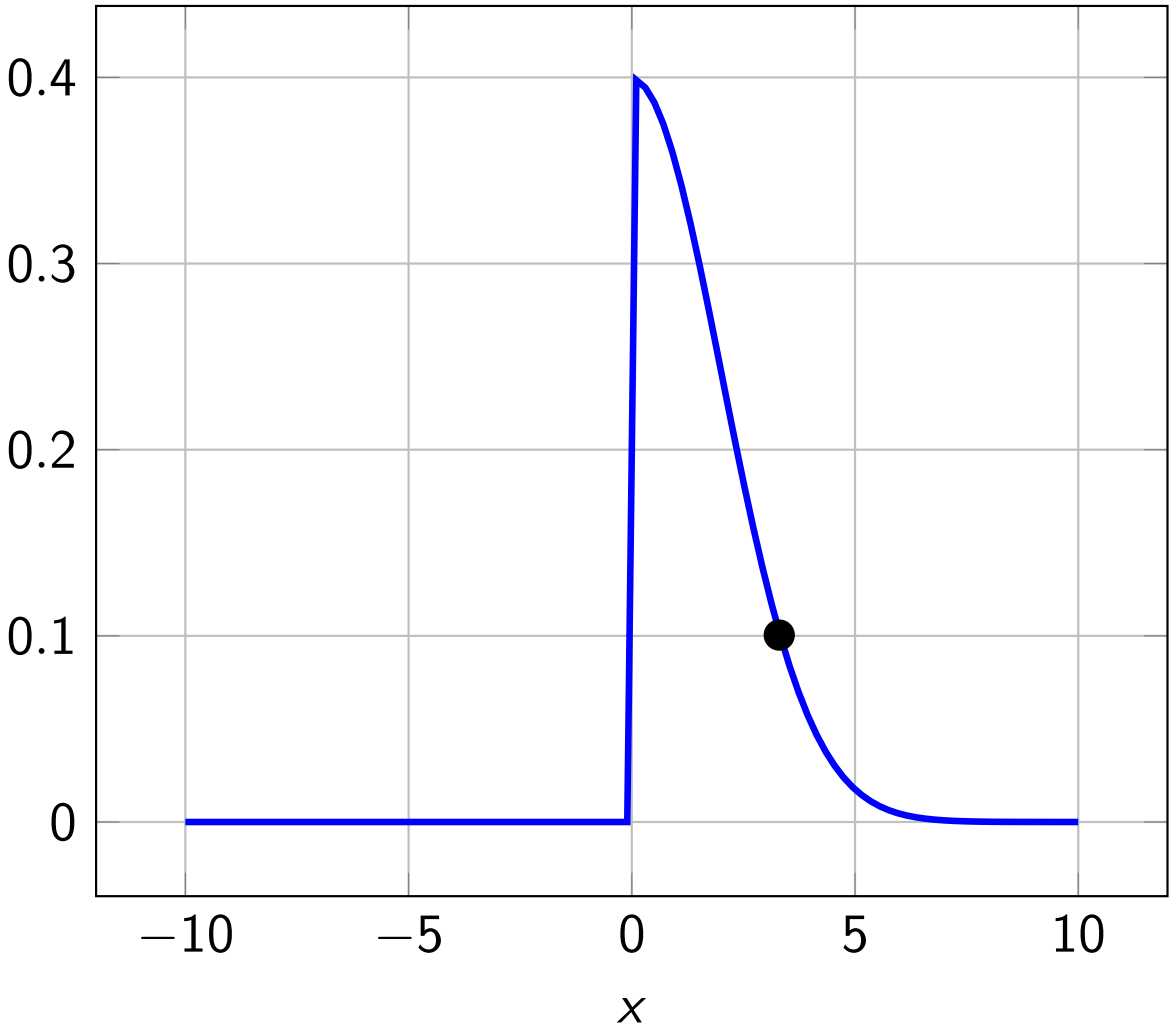
University of Nebraska Lincoln

What's going on here?



Input values $x < 0$ are transformed.
Their mass in the input distribution is shifted to $-x$ and accumulates in the output distribution

Here is the output distribution



This briefing is about

Probabilistic (Program Analysis)

as opposed to

(Probabilistic Program) Analysis

Why are you talking about this?

Why are you talking about this?

Modeling of common duckweed and aphid system

Why are you talking about this?

Modeling of common duckweed and aphid system

Assuring the safety of UAV operations in close proximity agriculture

Why are you talking about this?

Modeling of common duckweed and aphid system

Assuring the safety of UAV operations in close proximity agriculture

These systems involve inherent **uncertainty**

- in the input values, sensor error, state transitions, ...

Unlike in classic program analyses, this uncertainty can be characterized and should be exploited

University of Nebraska Lincoln

What kinds of questions can you answer?

How reliable is the program under an input distribution?

How frequently is this block executed?

Are these programs equivalent under an input distribution?

How sensitive is a test oracle to input distribution?

How much of the input space is covered?

How important is this dependence? (quantified info. flow)

If you can't prove it, how close did you get?

University of Nebraska Lincoln

What's next ...

- Provide an overview of the key concepts in data flow analysis and symbolic execution
- Describe how researchers have enriched those frameworks with probabilistic reasoning (of varying sorts)
- Break down the literature across 3 orthogonal dimensions
- Talk in more detail about probabilistic symbolic execution
- Give you a list of 5+ PhD topics in this area

University of Nebraska Lincoln