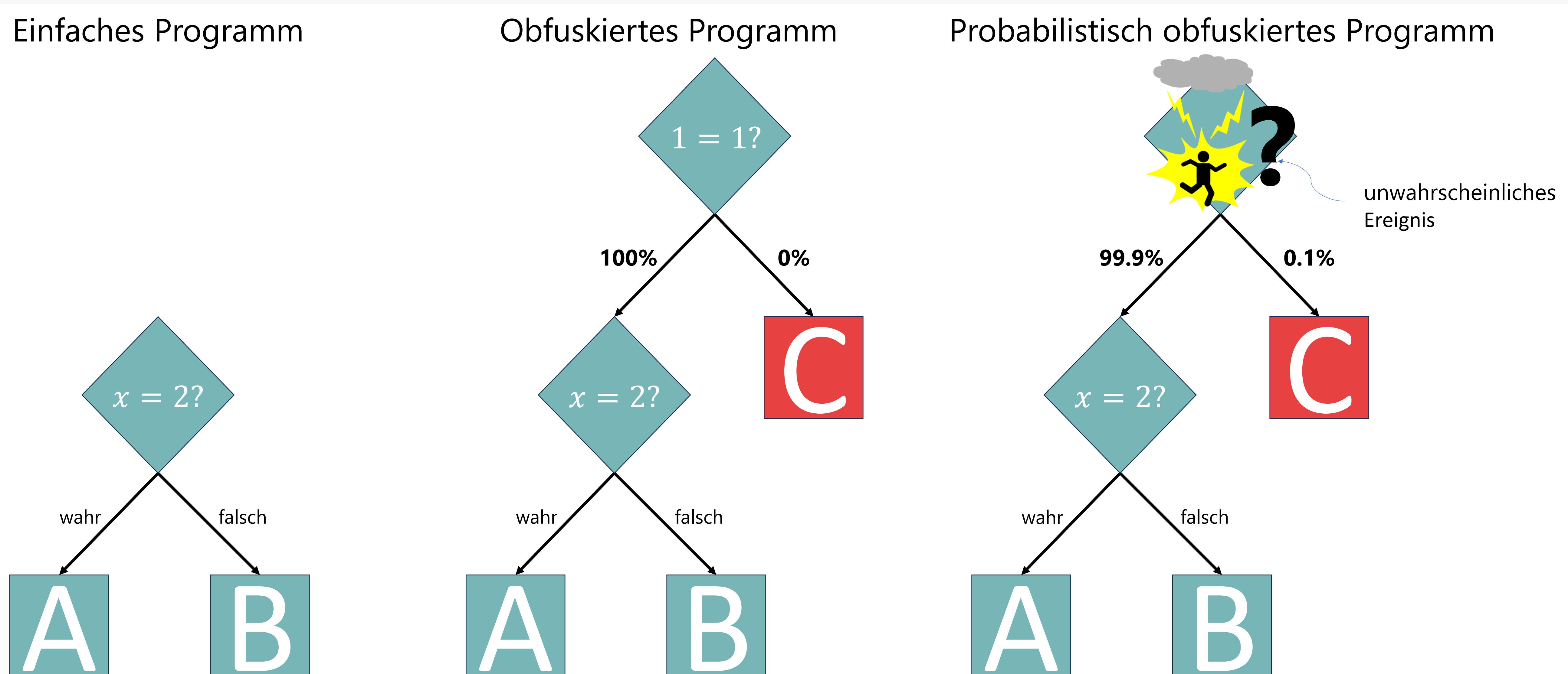


# Wie sicher ist Ihre Software wirklich?

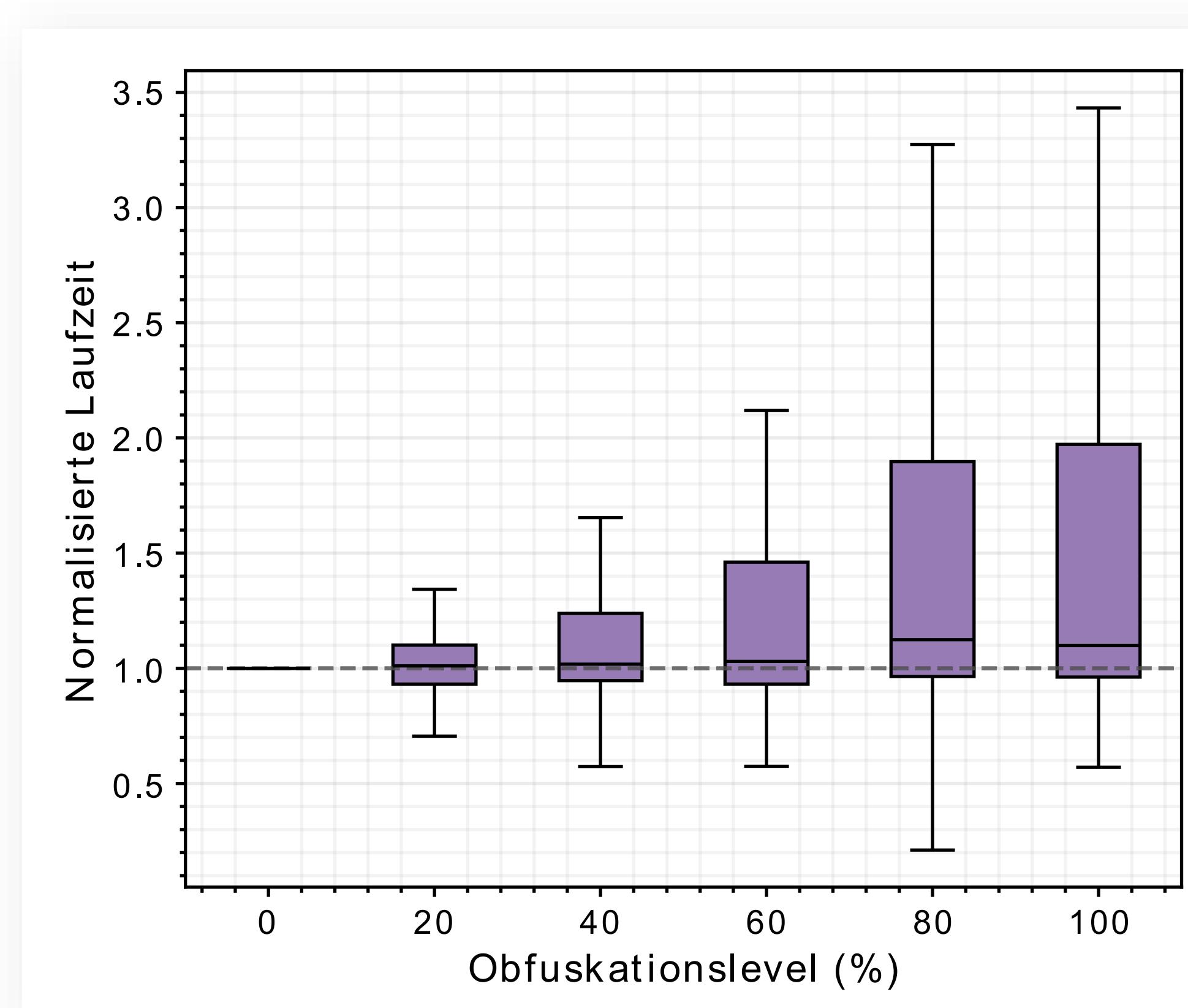
PoP: Probabilistische opake Prädikate gegen symbolische Ausführung

## Problem:

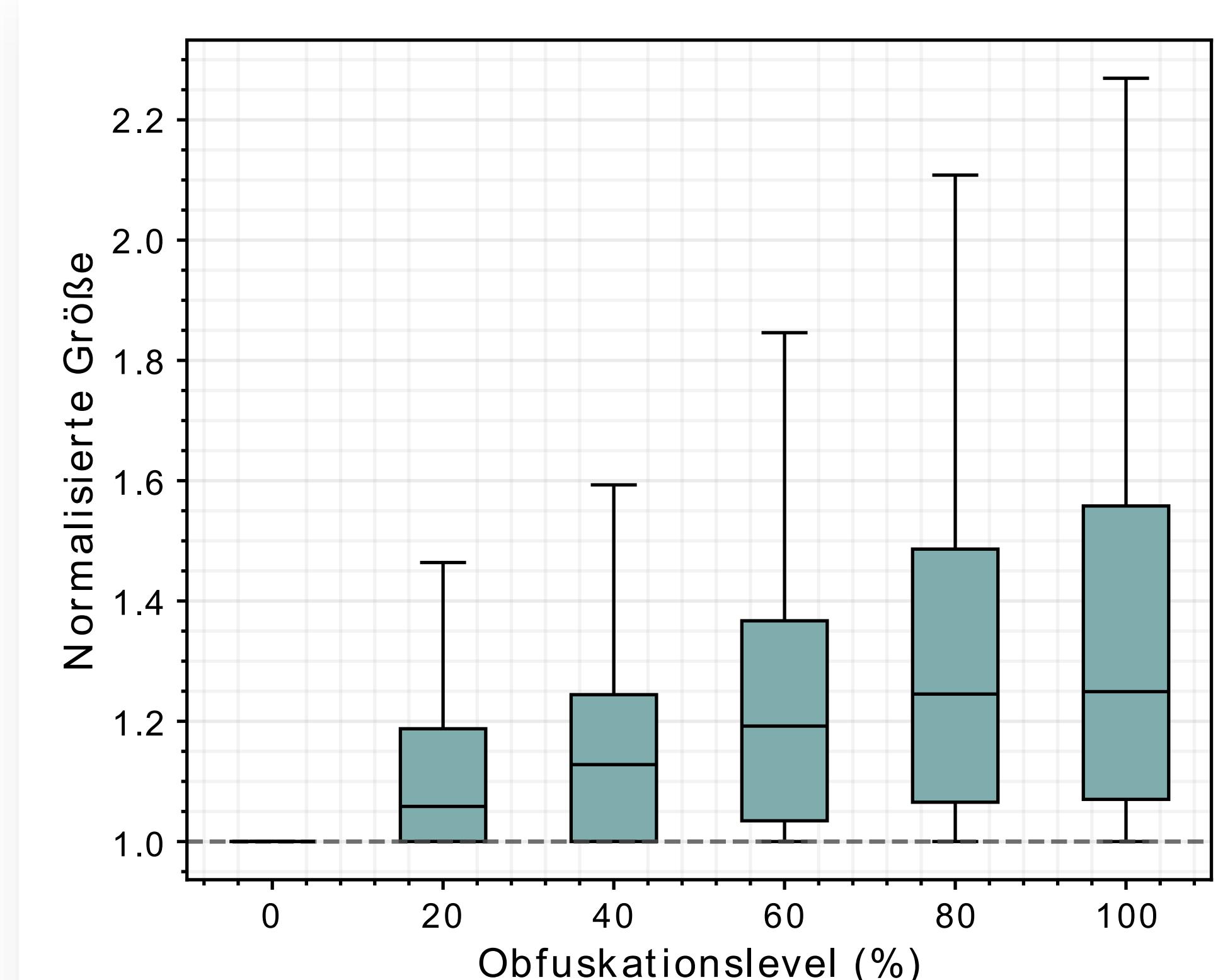
- Aktuelle Softwareverschleierungsmethoden (opake Prädikate) sind nicht resistent gegenüber manchen Angriffsmethoden (symbolischer Ausführung) → Möglichkeit, Software zu verstehen & zu rekonstruieren
- ~15 Mrd. Euro an Umsatz gehen jährlich in der EU durch Reverse Engineering/Cracking verloren [1]



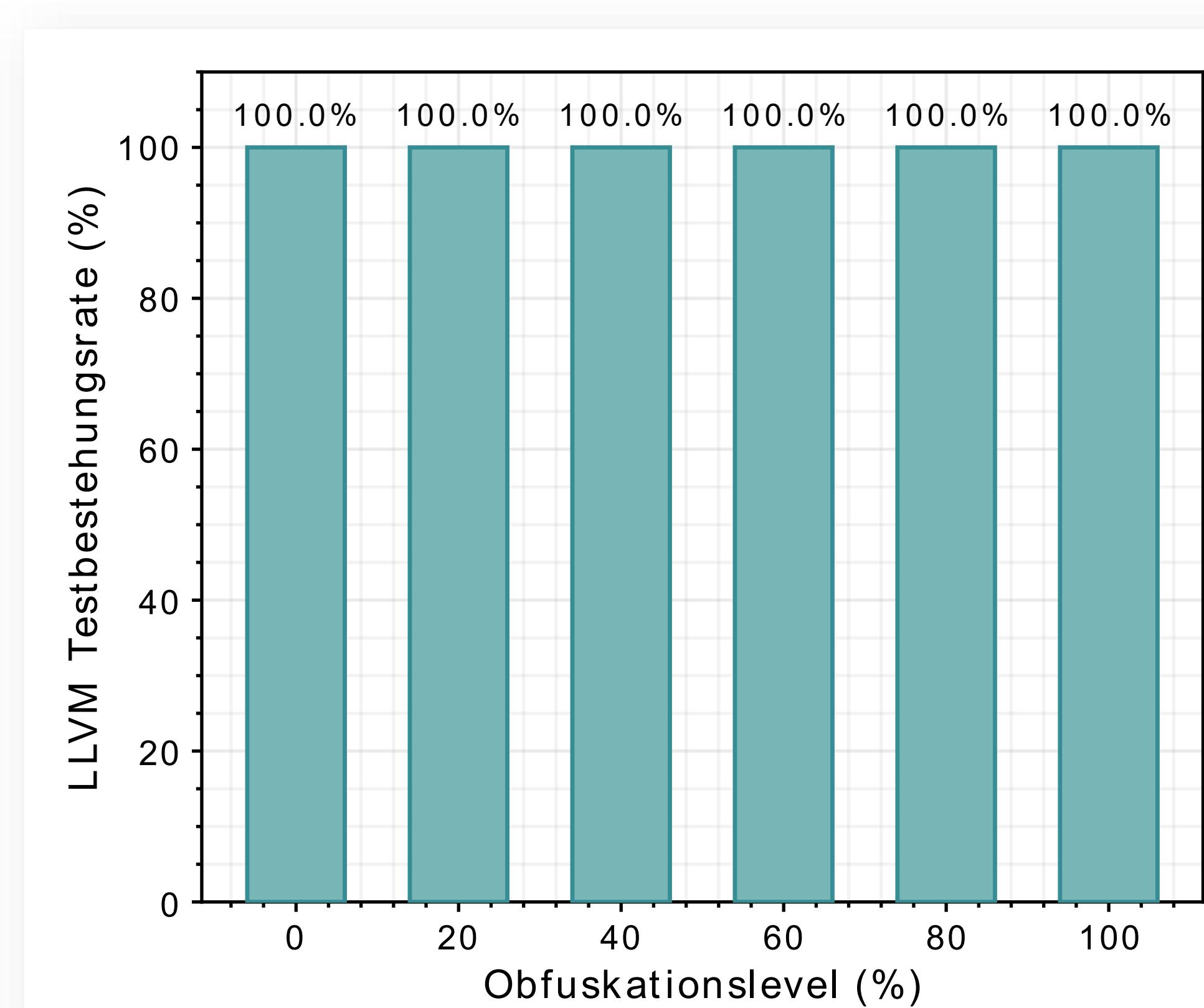
**Ergebnis 1:** Der **mediane Laufzeitimpakt** der neuen Obfuscation ist **gering**. Die Varianz ist allerdings groß.



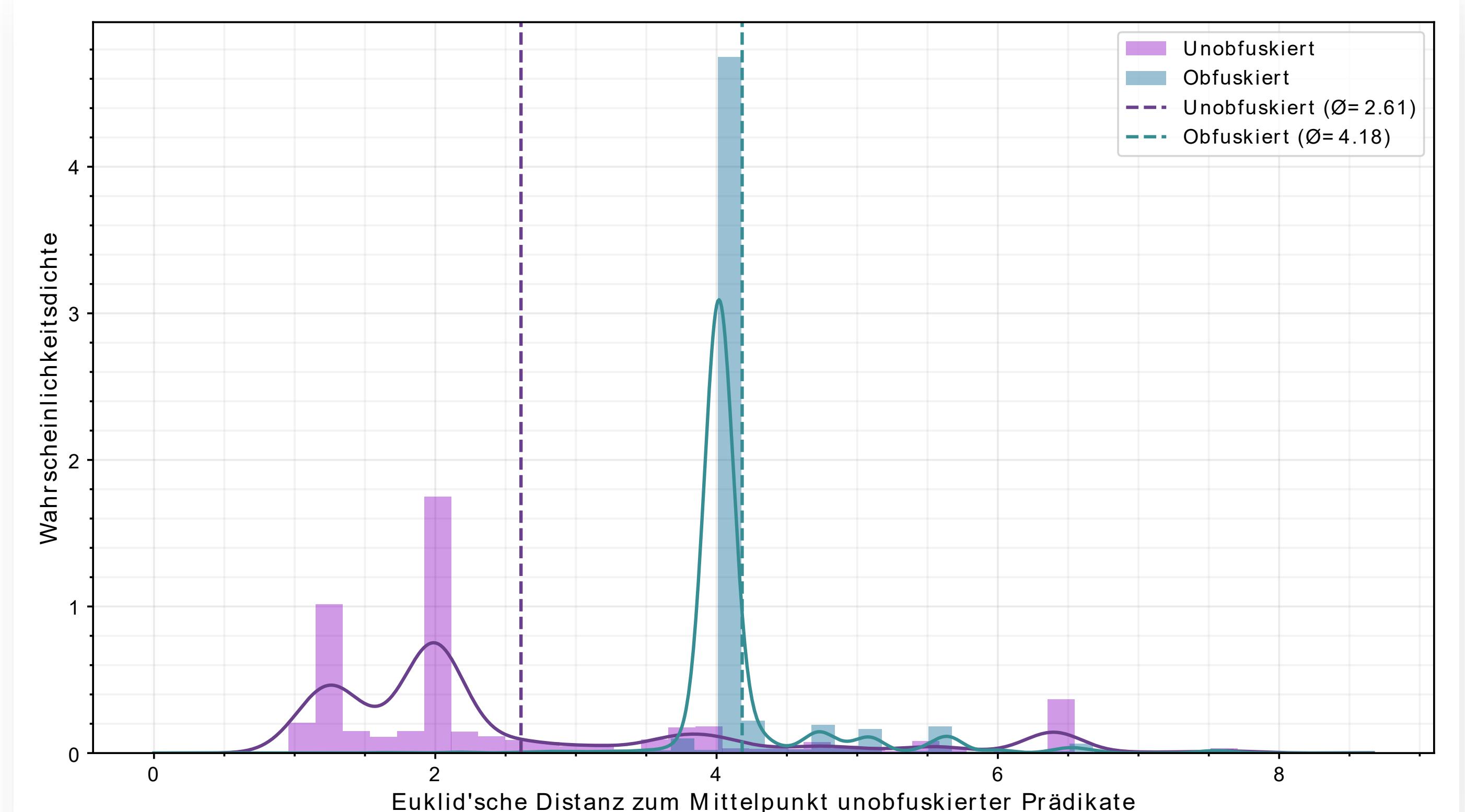
**Ergebnis 3:** Der **mediane Speicherimpakt** der neuen Obfuscation ist **gering**. Die Varianz ist allerdings groß.



**Ergebnis 2:** Trotz der geringen Wahrscheinlichkeit, dass die PoP-Prädikate den falschen Wert annehmen, **funktionieren alle Programme**.



**Ergebnis 4:** Der **eingefügte Programmcode unterscheidet sich statistisch** von regulären Programmen.

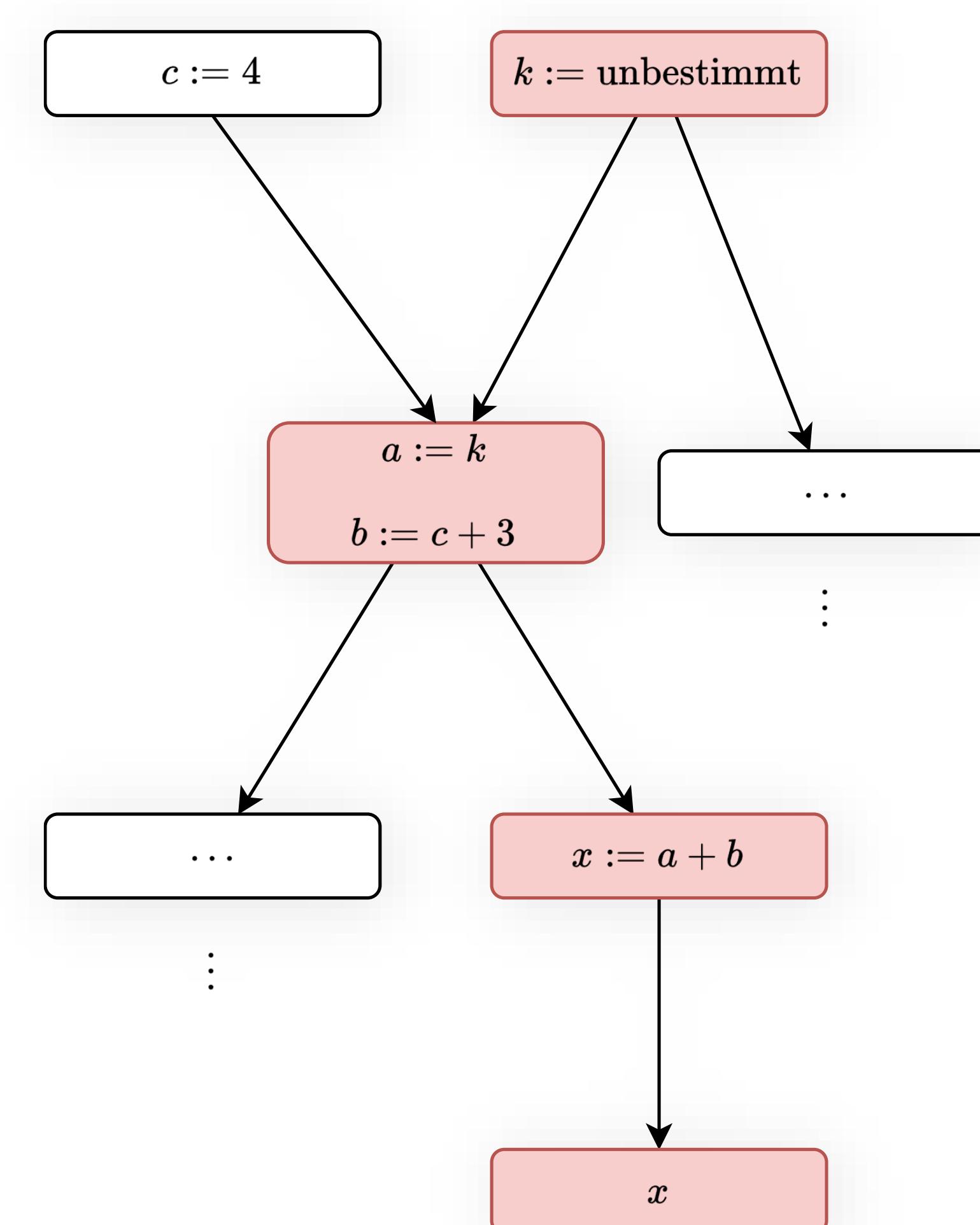


# Wie sicher ist Ihre Software wirklich?

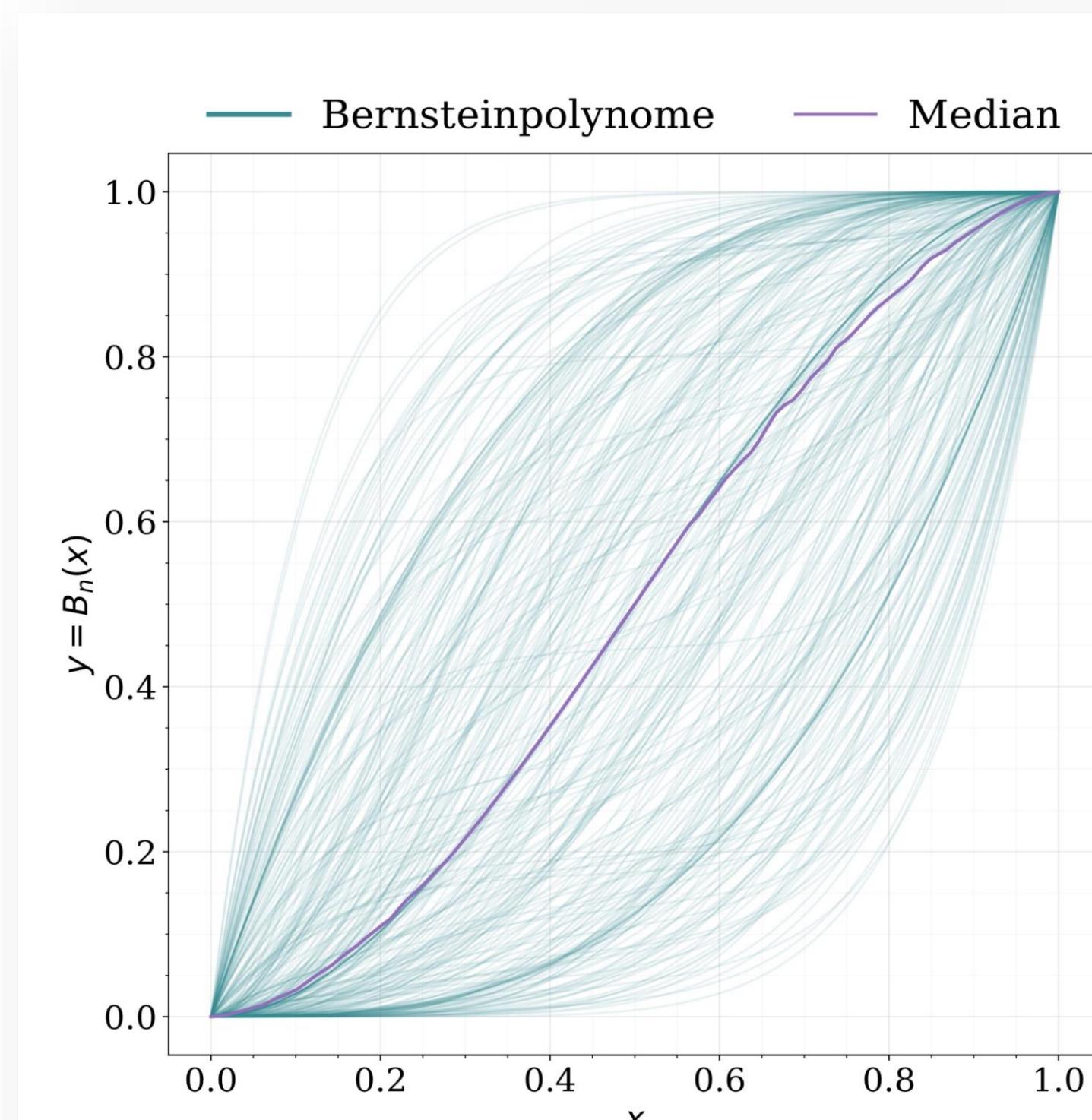
PoP: Probabilistische opake Prädikate gegen symbolische Ausführung

## Funktionsweise

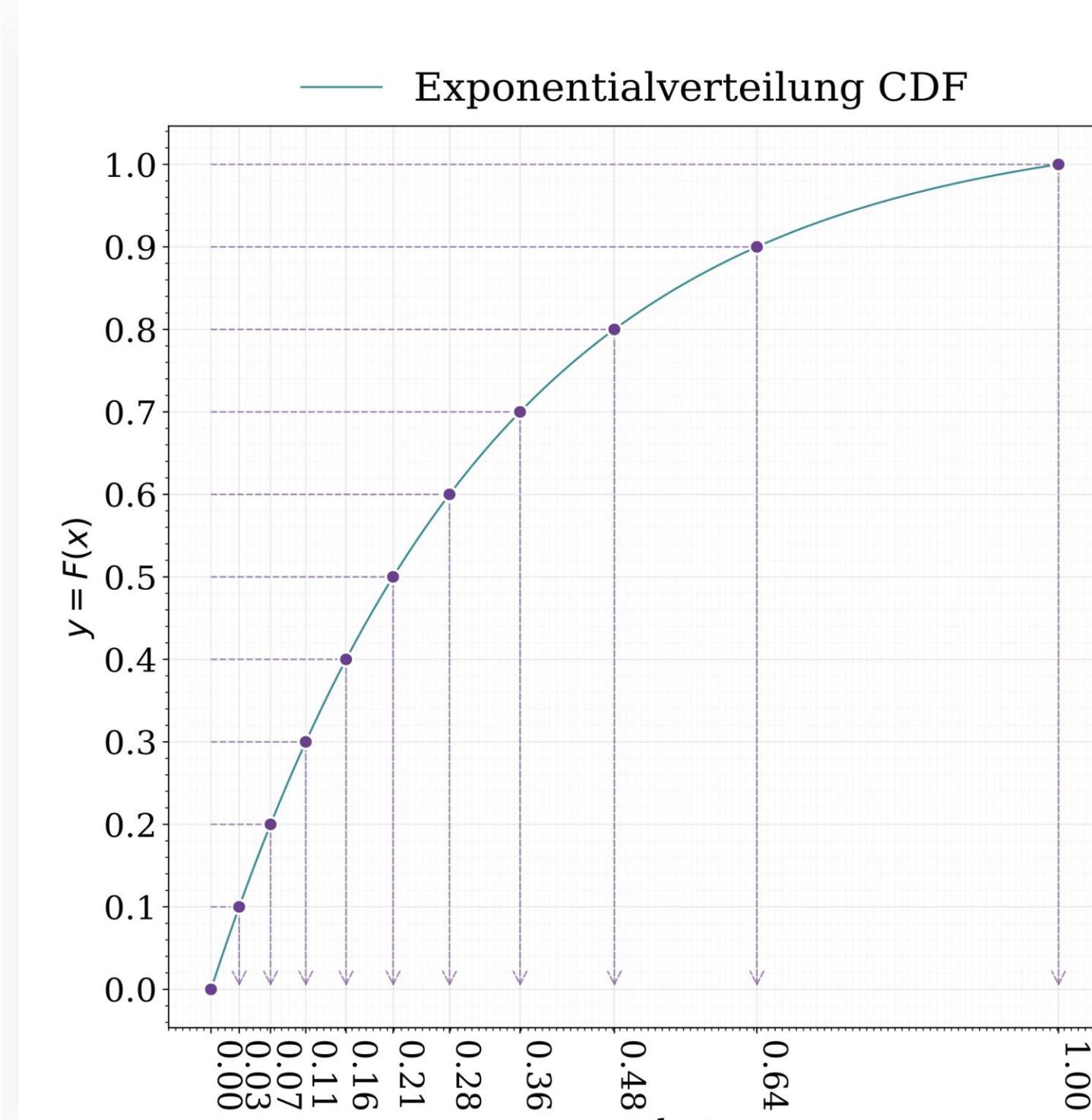
1. Parameter wählen, dessen Werte sich nicht statisch bestimmen lassen.



2. Kumulative Verteilungsfunktion generieren.



3. Über die Inversionsmethode Parameter gemäß der zugeordneten Wahrscheinlichkeitsdichtefunktion verteilen.



4. Da wir Kontrolle über die Verteilung der Zufallsvariable haben, können wir festlegen, dass bestimmte Intervalle/Werte besonders wahrscheinlich sind.

