

# *Kartézské genetické programování v jazyce Python*

Petr Dvořáček

# Motivace

## Kde lze využít CGP?

- návrh a optimalizace kombinačních obvodů
- návrh číslicových filtrů
- řešení problému symbolické regrese

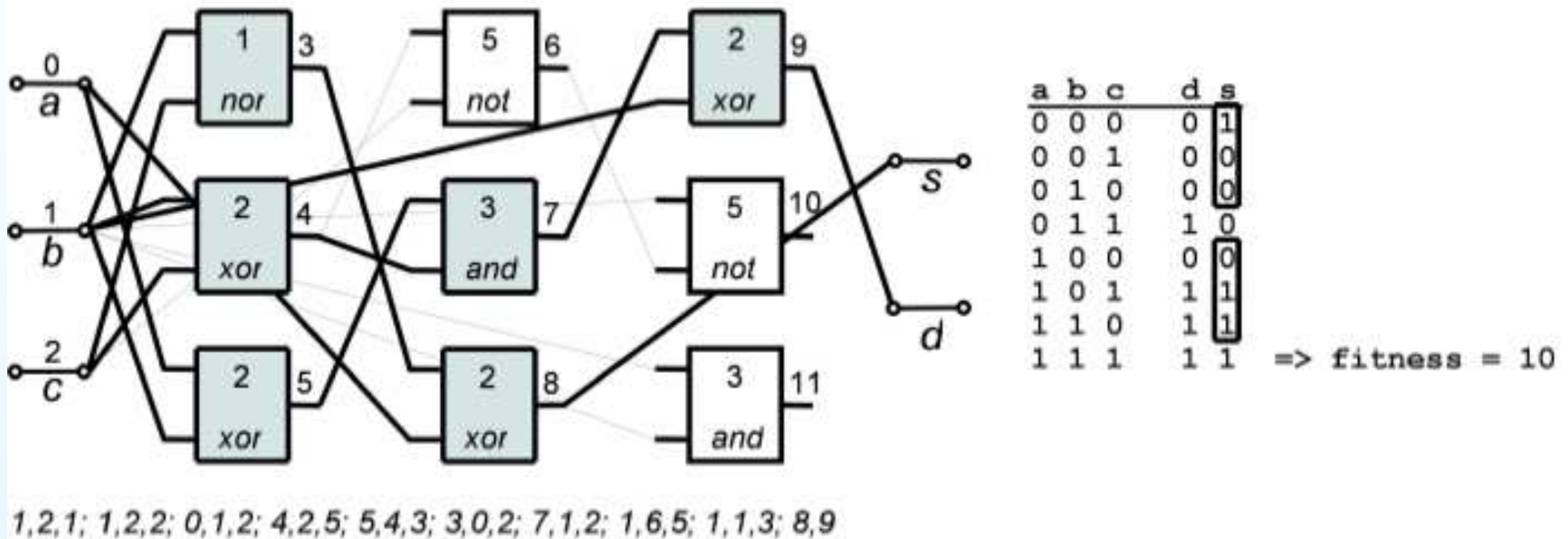
## Proč v Pythonu?

- rychlý vývoj programů
- ukázat možnosti akcelerace v Pythonu

## Proč akcelarovat CGP?

- evaluace fitness hodnoty je výpočetně náročná

# Princip výpočtu fitness hodnoty



- trénovací data prochází několikrát stejným chromozómem
- operace uzlu v kódu obsahuje hodně podmínek (switch case)

# Navržené akcelerační techniky

---

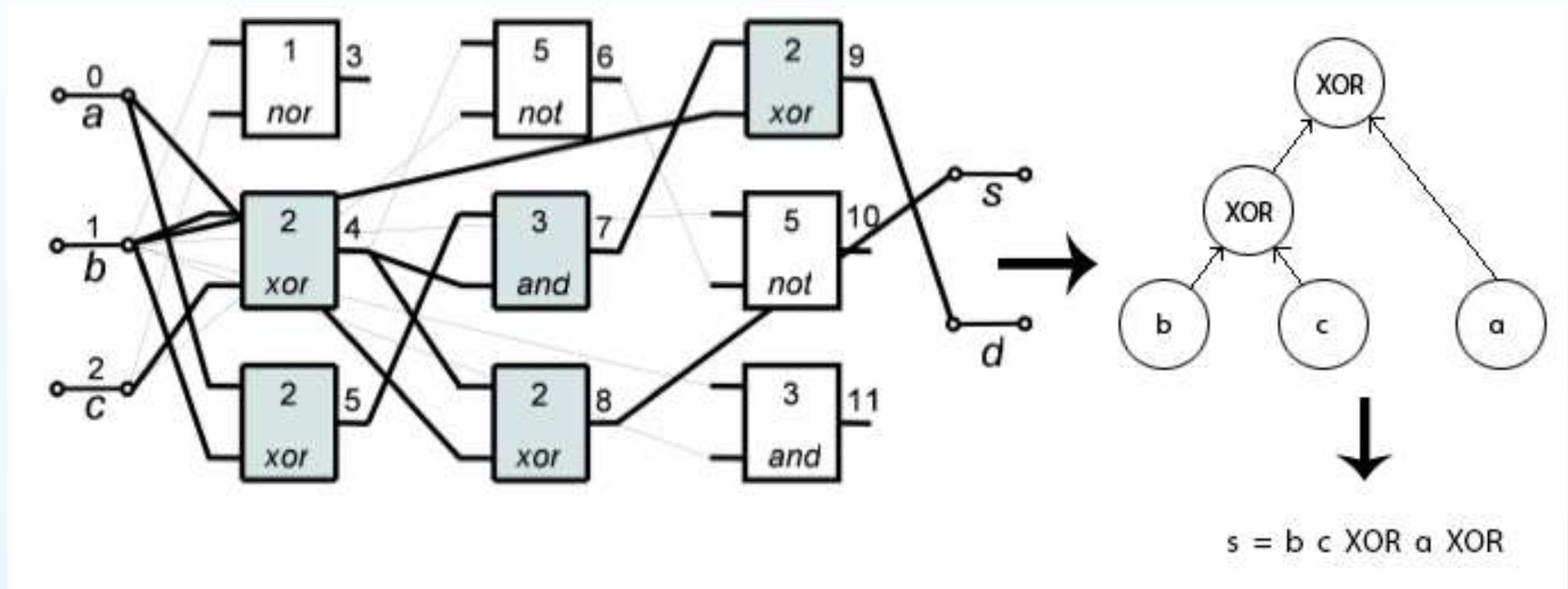
## Využití nástrojů pro akceleraci v Pythonu

- PyPy
- Cython

## Vyhodnocení fitness funkce

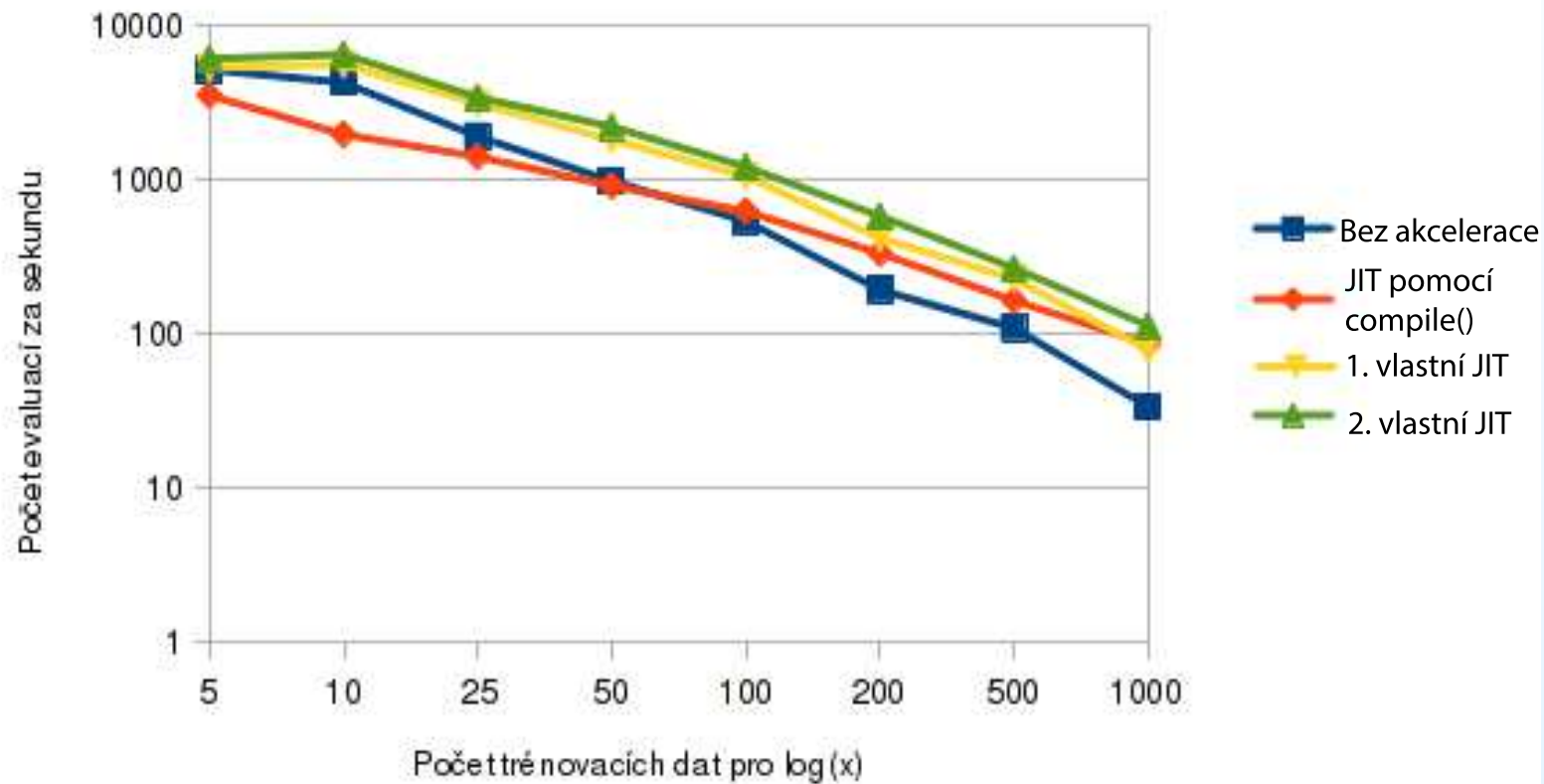
- pomocí paralelní simulace chromozómu
- just-in-time (JIT) kompilace chromozómu
  - pomocí vestavěné funkce `compile()`
  - vlastní kompilace chromozómu

# Vlastní JIT kompilace chromozómu

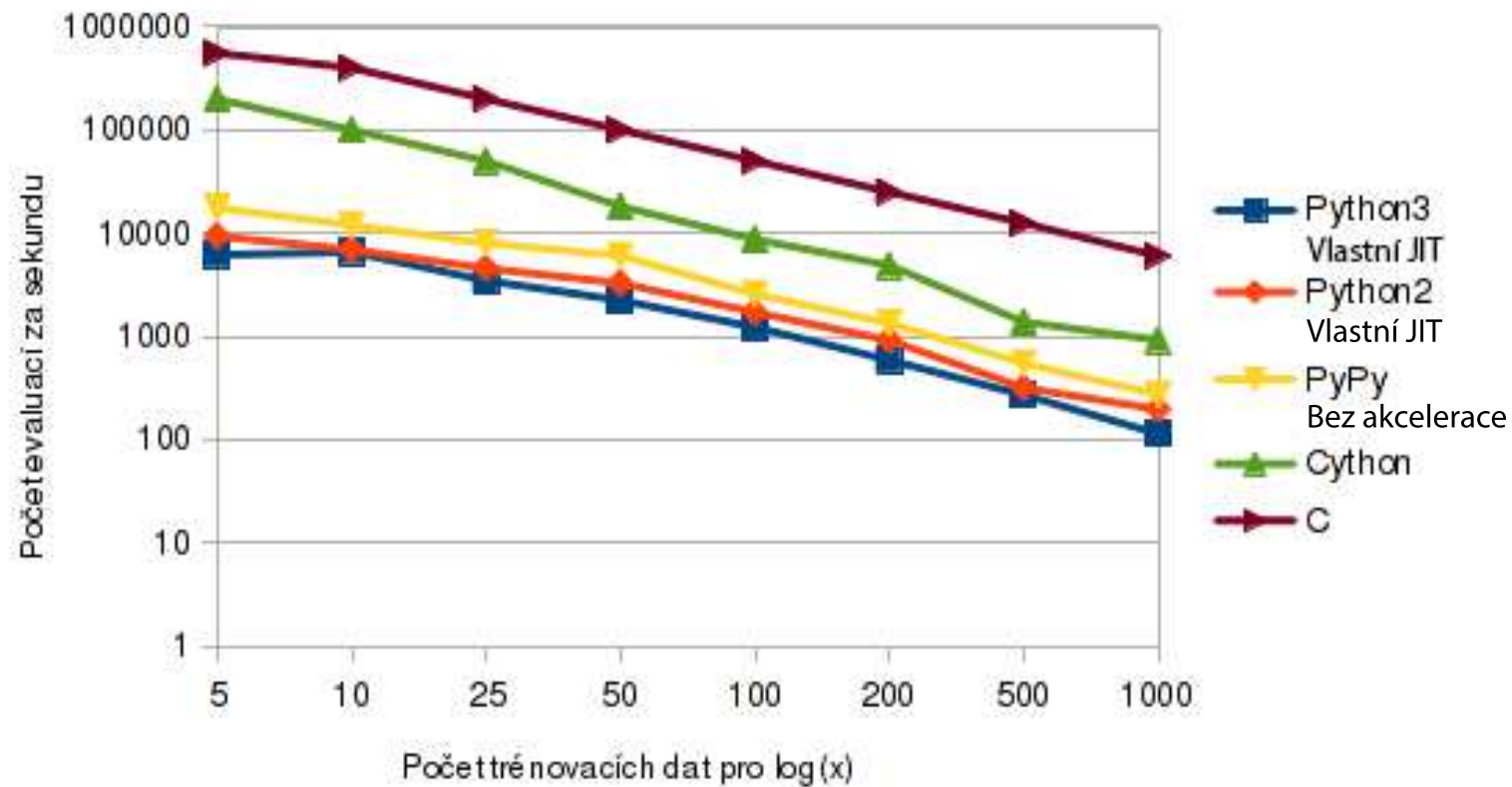


Fenotyp (použité uzly grafu) se převede do postfixové notace tak, aby se využila efektivita zásobníkové architektury.

# Výsledky



# Výsledky



# Závěr

Výpočet fitness pomocí JIT kompilace byl přibližně 4x rychlejší než původní algoritmus.

Byl vytvořen modul v jazyce Cython.

Další rozšíření:

- JIT kompilace chromozómu v jazyce C -> redukce skokových instrukcí
- vytvoření GUI
- přidání nových funkčních množin