

PyGDB — analiza strukture izvornog koda

Projekt iz kolegija Napredne baze podataka

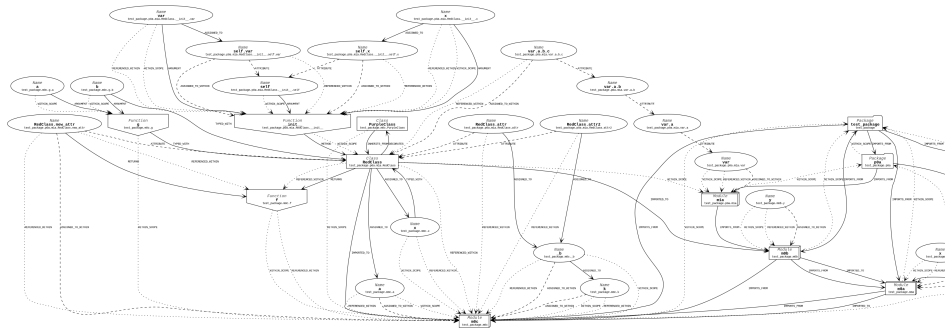
Luka Šimek

Prirodoslovno-matematički fakultet — Matematički odsjek
Sveučilište u Zagrebu

8. lipnja 2024.

- 1 Motivacija
- 2 Pokazna baza
- 3 Osvrt

- vrhovi: paket, modul, klasa, funkcija, ime
- odnosi: uvoz, nasljeđivanje, nazivni prostori, pridruživanja, atribut, metoda, tip...
- putevi: hijerarhije datoteka, klasa, opsega, uvoza
- upiti: filtriranje nepregledne količine podataka
- \Rightarrow grafovska baza podataka



Slika: Graf testnog paketa (60-tak linija ukupno)

Unosimo u bazu grafove 5 paketa:

- ovaj projekt,
- humanize,
- Requests,
- Seaborn,
- PyMC

...oko 30K vrhova i 220K bridova

- sve rekurzivne funkcije

```
MATCH (f: Function)-[b: RETURNS]->(f)
RETURN f, b;
```

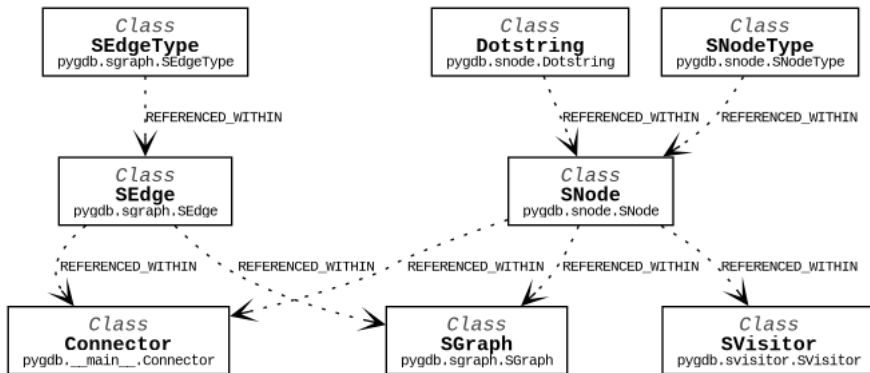
- nasljeđivanje u paketu Requests

```
MATCH (p: Package {fullname: 'requests'})
MATCH (p)<-[: WITHIN_SCOPE*]-(c: Class)
MATCH (c)-[b: INHERITS_FROM*1..]->(d: Class)
RETURN c, b, d;
```

- klase unutar ovog projekta i kad se „spominju”

```
MATCH (p: Package {fullname: 'pygdb'})
MATCH (p)<-[: WITHIN_SCOPE*]-(c: Class)
MATCH (c)-[b: REFERENCED_WITHIN]->(d: Class)
RETURN c, b, d;
```

Primjer rezultata



Slika: Rezultat trećeg upita

Statička analiza u dinamičkom jeziku:

- dinamički tipovi
- dinamički generirani kod (`exec`, `setattr`, `modul importlib...`)
- FFI

