

Zastosowanie analizy statycznej języka Ruby w edytorach tekstu

Rafał Łasocha

Promotor: prof. Witold Charatonik
Wydział Matematyki i Informatyki
Instytut Informatyki

13 grudnia 2018

- Interpretowany (interpreter i biblioteka standardowa w C)
- Dynamicznie typowany
- Zaawansowane mechanizmy obiektowe (wielokrotne dziedziczenie, klasy singletonowe)
- Metaprogramowanie (możliwość definiowania nowych klas, metod, zmiennych instancji, stałych w czasie działania programu)

- Informacja o “typie” wyrażenia
- Skok do definicji klasy lub metody
- Autouzupełnianie dostępnych metod

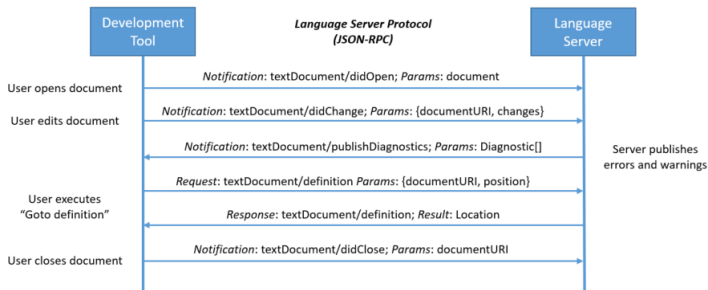
- Informacja o “typie” wyrażenia
- Skok do definicji klasy lub metody
- Autouzupełnianie dostępnych metod

Przykładowy kod

```
some_object.push(42)
```

Architektura projektu

- Protokół komunikacji z edytorem tekstu (*Language Server Protocol*)
- Silnik dostarczający funkcjonalności niezależny od edytora



Źródło: <https://microsoft.github.io/language-server-protocol/overview>

- 1 Parsowanie
- 2 Budowanie grafu DFG (*Data Flow Graph*)
- 3 Obliczenie typu dla każdego wierzchołka w grafie

Parsowanie

Nieciekawe.

Parsowanie

Nieciekawe.

Budowanie grafu DFG (*Data Flow Graph*)

Wymagane heurystyki, które obsługują specjalne przypadki takie jak wywołania metod `private` czy `attr_reader`.

Parsowanie

Nieciekawe.

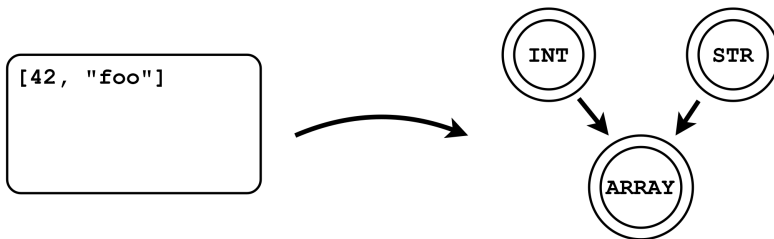
Budowanie grafu DFG (*Data Flow Graph*)

Wymagane heurystyki, które obsługują specjalne przypadki takie jak wywołania metod `private` czy `attr_reader`.

Obliczenie typu dla każdego wierzchołka w grafie

Obliczenie typu wierzchołka X zależy od aktualnie obliczonych typów wierzchołków z krawędzią do X . Obsługa typów parametryzowanych takich jak `Array<Integer or String>`.

Przykład



- obsługa metaprogramowania (w etapie zbierania definicji)
- budowanie inkrementacyjne
- integracja z dokumentacją YARD
- zbieranie informacji o niektórych efektach (w szczególności wyjątkach)