

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Teze diplomové práce

Portace knihovny Prawn z Ruby do Pythonu 3

Bc. Michal Molhanec

# Portace knihovny Prawn z Ruby do Pythonu 3

## Souhrn

Práce se zabývá automatickým překladem kódu napsaného v jazyku Ruby do Pythonu 3. Nejprve jsou prozkoumány principy překladu, jak je popisuje odborná literatura. V další části se porovnávají jednotlivé jazykové konstrukce a jsou navrženy způsoby jejich překladu.

Na základě této teorie byl vyvinut překladač, který svou funkčnost ukazuje na překladu knihovny Prawn pro generování PDF souborů.

## Klíčová slova

skriptovací jazyky, python, ruby, překlad, datové typy, pdf, abstraktní syntaktický strom

## Cíl práce a metodika

Základním cílem této práce je prozkoumání možnosti překladu kódu napsaného v jazyce Ruby do Pythonu 3. Ke konkrétní demonstraci je použita knihovna Prawn napsaná v Ruby, která slouží pro generování PDF souborů.

Vlastní portace bude založena na poloautomatickém překladu. Bude vyvinut automatický překladač, který zvládá podmnožinu jazyka Ruby přeložit do Pythonu. Jednodušší konstrukce budou překládány automaticky z Ruby AST do Pythonu, složitější konstrukce budou přepsány ručně.

## Teoretická východiska

Práce analyzuje možnosti překladu jazyků tak, jak je popisuje odborná literatura (Parr, 2010; Fowler, 2011).

Parr dělí překladače<sup>1</sup> v zásadě podle toho, jestli potřebují explicitní *sémantický model* (Fowler, 2011, s. 159) a jakým způsobem s ním pracují (Parr, 2010, s. 280):

- Překlad řízený vstupní syntaxí.
- Překlad řízený pravidly.
- Překlad řízený modelem.

Dále lze překladače dělit podle toho, jakým způsobem generují cílový jazyk. Například Fowler (Fowler, 2011, část VI):

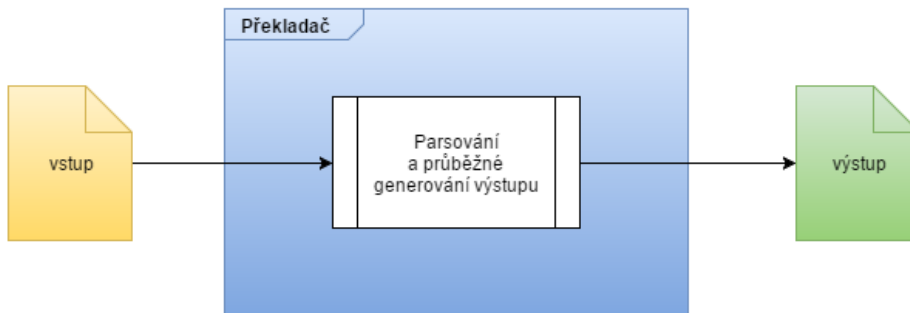
- *transformační převodník* (Fowler, 2011, s. 533), který prochází *sémantický model* a generuje zdrojový kód a
- vyplňování šablon (Fowler, 2011, s. 539).

---

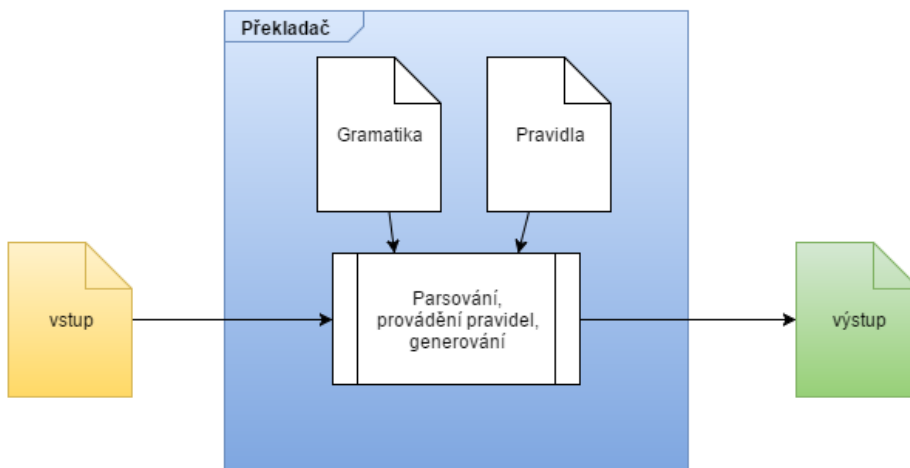
<sup>1</sup>V textu používám termín *překladač* pro program překládající jeden programovací jazyk do jiného.

Obrázek 1: Typy překladačů. Upraveno podle (Parr, 2010, s. 283-5)

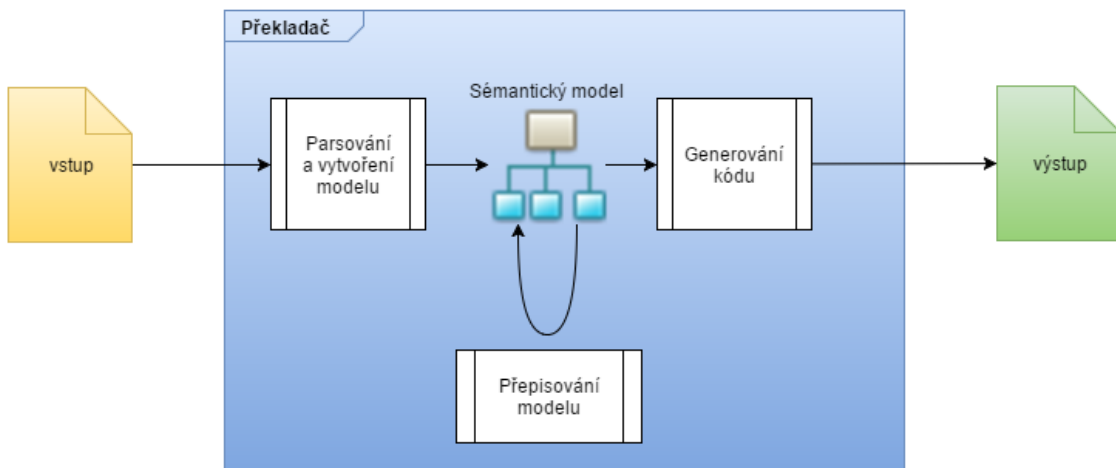
### Překlad řízený vstupní syntaxí



### Překlad řízený pravidly



### Překlad řízený modelem



# Porovnání vybraných konstrukcí jazyků Ruby a Python 3 a návrh mechanismu jejich překladač

Největší část práce zabírá porovnání jednotlivých konstrukcí a návrhu toho, jakým způsobem je přeložit z Ruby do Pythonu. Naznačme si problém na příkladu. Následující kód vypočte druhé mocniny čísel zadaných v poli a vrátí je opět jako pole:

```
# Ruby                                     # Python
čtverce = [1, 2, 3].map() do              def _blok(hodnota):
  |hodnota|                                return hodnota**2
  hodnota**2                               čtverce = list(map(_blok, [1, 2, 3]))
end
```

Levá strana obsahuje vstupní kód v Ruby, pravá pak jeden z možných ekvivalentů v Pythonu. Všimněme si následujících rozdílů:

- Python nemá přímý ekvivalent bloků, tedy anonymních klauzur. Musíme je tedy převést na jinou konstrukci. Protože tělo bloku tvoří jediný výraz, šlo by použít *lambda funkci*. My jsme použili standardní funkci, což je obecnější řešení.
- Není-li určena explicitně, je v Ruby hodnotou bloku (ale i metody) jednoduše hodnota posledního vykonaného výrazu. V Pythonu jsme naproti tomu museli uvést explicitní `return`, jinak by hodnotou funkce `_blok` byla prázdná hodnota (`None`).
- V Ruby je `map()` metoda pole, v Pythonu jde o globální funkci.
- V Ruby vrací `map()` pole, v Pythonu tzv. iterátor. Chceme-li získat pole, musíme ho vytvořit konstruktorem `list()`.

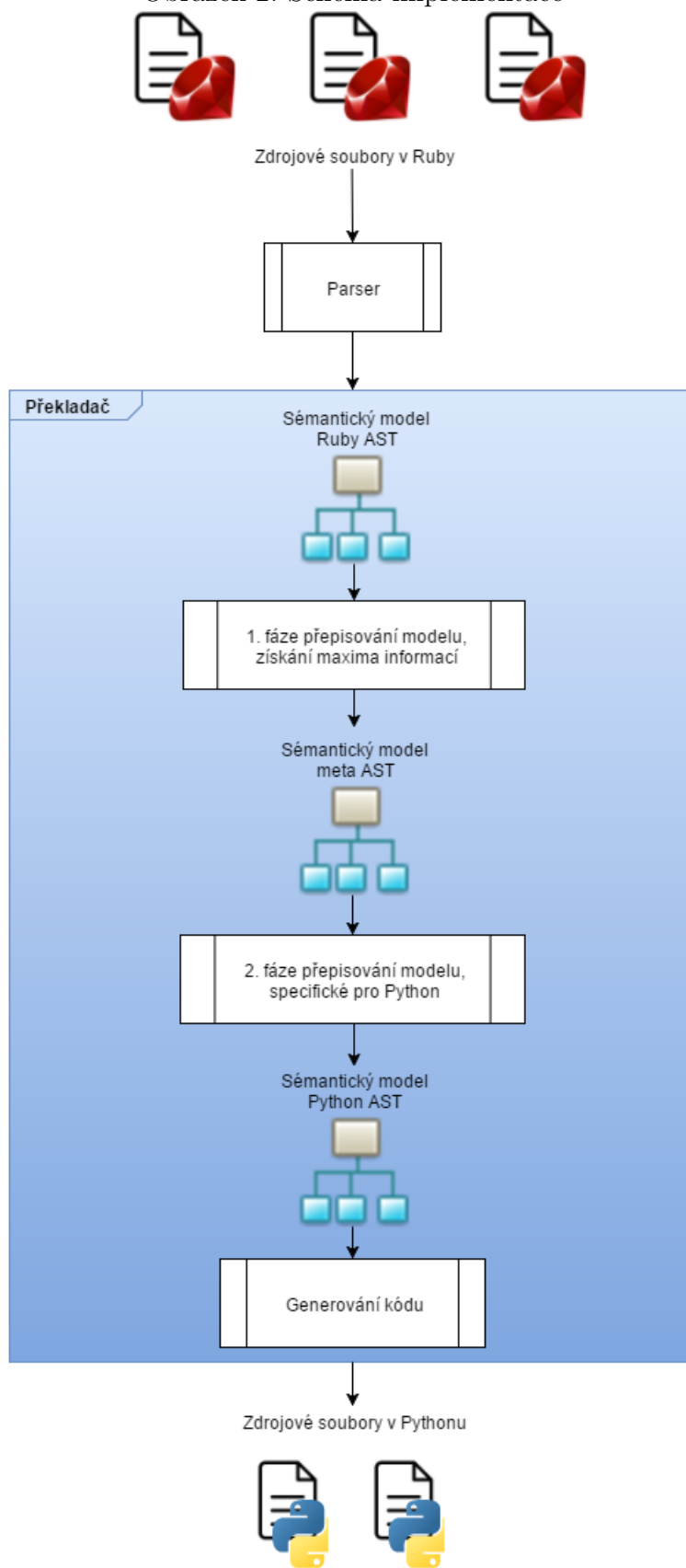
Vlastní kapitola se podrobně věnuje těmto, a mnoha dalším konstrukcím, a možnostem jejich překladač.

## Vlastní řešení

Vlastní překladač se dá rozdělit do několika fází, které ilustruje obrázek. Modrá plocha znázorňuje mnou implementovaný překladač.

- Vstupem je kolekce zdrojových souborů napsaných v jazyku Ruby.
- Ty jsou zpracovány parserem. Jak již bylo zmíněno v úvodních částech práce, vlastní parser jsem si nepsal, neboť parsery pro Ruby jsou k dispozici.
- Tím ze zdrojových souborů získáme jednotlivé syntaktické stromy. Nyní začne první fáze překladač. V té se snažíme sémantický strom odpovídající jazyku Ruby transformovat na univerzální model, „meta AST“. Procházíme vstupní model a snažíme se z něho získat maximum informací.

Obrázek 2: Schéma implementace



- Tím získáme meta AST. Meta AST by nemělo být závislé na Ruby, ani Pythonu. Aktuálně to není zcela splněno, je nicméně snaha oddělit alespoň transformace specifické pro Ruby (1. fáze) a specifické pro Python (2. fáze).
- Druhá fáze přepisování modelu realizuje právě transformace specifické pro Python. Zatímco v první fázi informace extrahujeme, v této fázi již může dojít i eliminaci informací, pokud nejsou pro generování Pythonu potřeba.
- Výsledkem je strom, který odpovídá výslednému zdrojovému kódu v Pythonu, tj. jeho následným parsováním bychom velmi teoreticky dostali znovu tento strom.
- Závěrečnou fází je generování. V ideální případě by mělo stačit projít výsledný model a generovat jednotlivé elementy. S tím, že šablony jednotlivých výsledných konstrukcí by již neměli obsahovat žádnou logiku, snad s výjimkou té úplně nejprostší. Aktuální implementace to ne zcela splňuje a bylo by dobré některé složitější šablony nahradit transformacemi, které by proběhly v druhé fázi.
- Výstupem jsou pak soubory v Pythonu tvořící balík (*package*) nebo modul (*module*). Umístíme-li je na cestu, kde je uvidí interpret Pythonu, pak by ho mělo jít z Pythonu běžně používat.

## Závěr

Cílem práce bylo porovnat možnosti jazyků Ruby a Python 3, vyvinout nástroj pro automatický překlad z Ruby do Pythonu a demonstrovat ho na překladu knihovny Prawn pro generování PDF souborů. To se ukázalo jako obří úkol, nicméně cílů se podařilo z větší části dosáhnout.

Přes polovinu této práce tvoří porovnání jednotlivých konstrukcí obou jazyků a možné varianty řešení jejich vzájemného převodu. I přes značný rozsah této části není jistě zcela vyčerpávající a bylo by možné na toto téma napsat ještě několikanásobně více. Přesto je popsána většina běžnějších konstrukcí, na které lze narazit při psaní „normálního“ kódu, tj. kódu, který se nesnaží využívat na maximum možnost metaprogramování.

Paralelně vznikl i překladač z Ruby do Pythonu 3. I přesto, že jsem si záměrně vybral jazyky, které k sobě mají relativně blízko, to byl nesnadný úkol. Kromě očekávaných problémů plynoucích ze základních rozdílů mezi jazyky (měnitelné řetězce v Ruby a neměnitelné v Pythonu, neexistence přímého ekvivalentu bloků v Pythonu, při neuvedení explicitního `return` je návratovou hodnotou metody hodnota posledního vykonaného výrazu v Ruby, `None` v Pythonu), se objevily i problémy překvapivější. Například fakt, že řetězce v Ruby mohou obsahovat libovolná binární data. Dále je to samozřejmě řada drobností, kdy se na první pohled zdánlivě podobné konstrukce chovají v detailech jinak.

Co se týče knihovny Prawn, bylo dosaženo stavu, kdy do Pythonu přeložená verze dokáže generovat jednoduché PDF dokumenty, které mohou obsahovat obrázky. Zatím zdaleka nebyla otestována celá její funkčnost. Také její použití z Pythonu zatím není příliš přirozené, je nutné používat speciální třídu pro řetězce a podobně.

Jako překvapivý bonus jsem objevil pár drobných chyb v knihovně Prawn. Tyto chyby prověřím a případně pošlu autorům knihovny Prawn.

## Seznam použitých zdrojů

FLANAGAN, David a MATSUMOTO, Yukihiro. 2008. *The Ruby programming language*. 1. vyd. O'Reilly. ISBN 978-0-596-51617-8.

FOWLER, Martin a PARSONS, Rebecca. 2011. *Domain-Specific Languages*. 1. vyd. Addison-Wesley. ISBN 03-217-1294-3.

PARR, Terence. 2010. *Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages*. 1. vyd. The Pragmatic Programmers. Pragmatic Bookshelf. ISBN 19-343-5645-X.

PERROTTA, Paolo. 2011. *Metaprogramming Ruby: Program Like the Ruby Pros*. 1. vyd. The Pragmatic Programmers. Pragmatic Bookshelf. ISBN 19-343-5647-6.

SHAUGHNESSY, Pat. 2014. *Ruby Under a Microscope: An Illustrated Guide to Ruby Internals*. 1. vyd. No Starch Press. ISBN 1-59327-527-7.