

Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce István Satmári
Název práce Frege IDE with JetBrains MPS
Rok odevzdání 2018
Studijní program Informatika **Studijní obor** Softwarové a datové inženýrství

Autor posudku Pavel Parížek **Role** Vedoucí
Pracoviště Katedra distribuovaných a spolehlivých systémů

Text posudku:

Cílem diplomové práce bylo vytvořit implementaci funkcionálního programovacího jazyka Frege v rámci platformy JetBrains MPS, a následně zhodnotit do jaké míry je tato platforma obecně vhodná pro funkcionální jazyky. Oba tyto cíle byly splněny ve velmi uspokojivé míře.

Rozsah práce, jak implementace tak i textové části, je výrazně nadprůměrný. Autor implementoval poměrně velkou podmnožinu jazyka Frege ve MPS. Zásadní část implementace, na kterou autor potřeboval nejvíc času, jsou transformační akce editoru, které velmi usnadňují psaní zdrojového kódu. Na základě osobní zkušenosti se MPS, kterou získal během řešení práce, autor provedl diskuzi obecné použitelnosti MPS na funkcionální jazyky. Součástí diskuze je popis výhod a omezení MPS v tomto kontextu, a také srovnání projekčních editorů s klasickými textovými editory. Text má vysokou úroveň, je srozumitelný a čtivý.

Dále přikládám hodnocení konzultanta ze JetBrains, kterým byl Václav Pech.

=====

Text práce je celkově srozumitelný a přehledně strukturovaný. Nástroj MPS je v práci podrobně popsán a všechny jeho důležité části a možnosti vysvětleny pro čtenáře bez předchozích zkušeností s MPS. Rovněž jazyk Frege a jeho odlišnosti od jazyka Haskell jsou popsány. Vlastní implementace Frege v MPS je zdokumentována srozumitelně a přehledně, byť poněkud povrchně, což čtenáře občas nutí nahlížet do kódu. Na závěr uvedeno dobře zpracované porovnání textových a projekčních editorů a možnosti aplikace projekčních editorů pro tvorbu IDE pro funkcionální jazyky.

Implementace jazyka je s ohledem na omezené časové možnosti diplomové práce poměrně vydařená a dostatečně obsáhlá, Struktura konceptů je propracovaná, byť místy nejspíše složitější, než bylo nutné. Editace většiny konceptů je plynulá a intuitivní. Škoda, že při editaci definic funkcí jsou někdy základní úkony poněkud problematické. Zdrojový kód je dobře napsaný, prospěla by mu větší přehlednost pro rychlejší orientaci v kódu. V sandboxu je připraveno velké množství příkladů ukazujících schopnosti Frege-IDE. Poslouží tedy i jako dobrý počáteční bod pro analýzu projektu.

K implementaci v bodech:

Klady

- Velmi důmyslně propracovaný systém modulů, importů a exportů
- Komplexní řešení scope a ScopeProviderů
- Vyjma typových tříd je možné definovat všechny důležité výrazové prostředky jazyka Frege.

Zápory

- Příliš mnoho konceptů
- Poněkud nepřehledné strukturování konceptů (např Guards - část je v Expression.Guards, část je ve FunctionDefinition)
- Mnoho věcí je načato, ale nedokončeno.
- Občas kryptická jména (např koncept FDGCEName)
- Občas nepřehledný, místy duplicitní kód s nejasnou funkcionalitou (Např FDASVarNameTransformationMenu_Default "=" téměř totožná s PatternArgument_SetFunctionDefinitionToAssignmentOrGuard "=", ale jedno transformační menu vytvoří FDAssignment a druhé FDGrouped)

K užívání jazyka v bodech:

Klady

- Vytvořeno celkem vysoké množství transformačních menu, při některých činnostech, například definice nového typu či anotování funkce, je možné být ve Frege-IDE i rychlejší než v textovém editoru.

Zápory

- Zdlouhavé vytváření funkcí, celkově není editování funkcí příliš plynulé
- Občas lze narazit na podivné chování (např. " foo + '=' " != "foo x + '=' " -> Pokaždé se vytvoří jiný koncept)

K typovému systému v bodech:

- Obstojně zvládá currying, tedy částečnou aplikaci funkcí
- Obsahuje zvláštní typ 'undecidable', který je nadtypem i podtypem všech typů, kterým autor modeluje chybějící typové proměnné. To ale nestačí k rozumnému otypování např. higher-order funkcí jako fmap, či jen obyčejné identity, ze které se místo forall a . a -> a stane něco výrazně slabšího jako forall a b . a -> b (místo zaručení stejného typu vstupu a výstupu povolí na obou koncích libovolný)
- Neobsahuje Typeclasses
- Neprovádí typovou kontrolu pro datatypy

Závěr

Mnoho navržených schopností nakonec zůstalo z časových důvodů neimplementováno. Vzhledem k přehlednosti projektu, kvalitě jeho základního návrhu a zvolené open-source licenci lze práci považovat za velmi zdařilý začátek, na který je možné snadno navázat a v projektu pokračovat. Možné směry dalšího rozvoje projektu jsou naznačeny v práci samotné - plná implementace typového systému, doplnění chybějících konceptů a plynulejší editace kódu v některých situacích.

Práci doporučuji k obhajobě.

Práci nenavrhuji na zvláštní ocenění.

Pokud práci navrhuje na zvláštní ocenění (cena děkana apod.), prosím uveďte zde stručné zdůvodnění (vzniklé publikace, významnost tématu, inovativnost práce apod.).

Datum 25.8.2018

Podpis