

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Jiří Klepl
Název práce Type inference and polymorphism for C
Rok odevzdání 2020
Studijní program Informatika
Studijní obor IOI

Autor posudku Miroslav Kratochvíl Vedoucí
Pracoviště Katedra softwarového inženýrství

K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání		X		
Splnění zadání		X		
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>		X		
<p>Cílem práce je do jazyka C naportovat polymorfismus a přetěžování funkcí známé z lambda kalkulu, konkrétně z jazyků rodiny ML a Haskellu. Cíl je dobře opodstatněný, autorovou hlavní motivací je umožnit jednodušší použití užitečných generických konstrukcí, které se v C buď nepoužívají, nebo jsou ‘implementovány’ preprocesorem.</p> <p>Práce cíl splňuje, výsledný program kompiluje jazyk C se syntaxí rozšířenou o parametrický polymorfismus (pojmenovaný ‘CHM’) do standardního C.</p>				

Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i>		X		
Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>		X	X	
Analýza		X		
Vývojová dokumentace		X		
Uživatelská dokumentace		X		

Anglicky psaná práce popisuje základy typových systémů, motivaci pro typové třídy, a odkazuje se na standardní ‘Typing Haskell In Haskell’ pro implementaci vlastního typového systému. Autor pak popisuje přepsání jazyka C do struktury THIH, nad kterou spouští typovou inferenci a výsledky používá k nahrazení generických konstrukcí konkrétními typovými konstantami (v procesu instanciaci). Struktura implementace i použití programu jsou dokumentované dostatečně.

Autor se vyhýbá některým problémům způsobeným nekompatibilitou jazyka C a použité typové teorie (např. konstantnosti a typovým třídám popisujícím obsah struktur), systematické vyřešení těchto problémů je ale mimo rámec práce.

Drobné problémy textové části jsou následující:

- Angličtina je sice dobře čitelná a většinou pochopitelná bez potíží, struktura textu by ale občas zasloužila víc pozornosti — především v úvodu práce, který moc dobře nepopisuje, co čtenáře v práci vlastně čeká. Některé věty jsou strukturou výrazně odlišné od očekávané ‘scientific english’, příčinu lze většinou najít v pokusu spojit příliš mnoho myšlenek do jedné věty. Stylistické potíže ale vadí jen okrajově.
- Některá fakta v textu by si zasloužila přesnější referenci na zdroj informace nebo lepší vysvětlení. Např., různá tvrzení o rozhodnutelnosti typových systémů, diskuze o (ne)podobnosti s C++ na konci sekce 3.4.1, popis ‘Record fields’ (§3.4.2, chybí příklad nebo odkaz na existující implementaci), tvrzení o nerozhodnutelnosti zastavení procesu instanciaci (§3.4.3), a odkaz na ‘issues with subtyping’ (§2.2.3).

Závěr práce a vyhodnocení chybějící funkcionality dává poměrně jasný pohled na potenciální budoucnost kompilátoru a navrhuje rozumná, dobře prozkoumaná a realizovatelná rozšíření. (Souhlasit nejde snad jen s navrhovaným systémem importovatelných modulů, který je v ‘systémovém’ jazyce z mnoha zákeřných příčin bohužel prakticky nerealizovatelný.)

Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu	... architektura, struktury a algoritmy, použité technologie	X			
Kvalita zpracování	... jmenné konvence, formátování, komentáře, testování		X		
Stabilita implementace			X		

Implementace využívá Haskellovou knihovnu ‘Language.C’ na parsování a emitování kódu; autor tuto knihovnu vhodně rozšiřuje o podporu syntaxe a anotací potřebných pro fungování CHM. Celková implementace je poměrně rozsáhlá, ‘centrální’ kód programu (bez úprav v jiných knihovnách a podpůrného ‘knihovního’ kódu) má přes 2500 řádků hodně zhuštěného Haskellu. Kód je vhodně rozdělený do samostatných knihoven.

Demonstrační programy jsou zajímavé a koncept a použití pokrývají dostatečně. Za zmínku stojí hlavně program s implementací generického vektoru a některé pěkné vlastnosti kódu, např. bezparametrové `new` a ‘překvapivá’ instanciaci celého vektoru z příkladu pro `typ char`.

Jedinou (zbytečnou) otázkou zůstává význam pojmenování standardní knihovny jako ‘STL’ — kolize názvu s C++ by nevadila, ale důvod pro písmeno T mi ve zkratce chybí (především potom, co autor šablony pracně nahradil jinou konstrukcí).

Celkové hodnocení Výborně

Práci navrhuji na zvláštní ocenění Ne

Datum

Podpis