

Posudek oponenta závěrečné práce

České vysoké učení technické v Praze

Fakulta informačních technologií

Student: Bc. Jan Noha
Oponent práce: Ing. Jan Trávníček
Název práce: LLVM frontend for the Scheme language
Obor: Systémové programování

Datum vytvoření: 18. 5. 2016

Hodnotící kritérium:	Způsob hodnocení - následující škálou 1 až 5:
1. Náročnost a další komentář k zadání	1=mimořádně náročné zadání, 2=náročnější zadání, 3=průměrně náročné zadání, 4=lehčí, ale ještě dostatečně náročné zadání, 5=nedostatečně náročné zadání
Popis kritéria: Podrobněji charakterizujte diplomovou (bakalářskou) práci a její případné návaznosti na předchozí nebo běžící projekty. Dále posuďte, čím je zadání této ZP náročné. (U obtížnější ZP lze dále tolerovat některé nedostatky, které by u ZP standardní obtížnosti tolerovány nebyly; a naopak u jednoduché ZP mohou být zjištěné nedostatky hodnoceny přísněji.)	
Komentář: Téma se zabývá novou implementací kompilátoru jazyka Scheme pomocí LLVM do nativního kódu. Samotná tvorba kompilátoru by neopodstatňovala volbu náročnějšího zadání. Scheme je ale funkcionální dynamicky typovaný jazyk, který podporuje také tak zvané first class functions. Výsledný spustitelný program musí být schopen kompilovat do nativního kódu i programově vytvořené zdrojové kódy nových funkcí, a ty pak volat.	
Hodnotící kritérium:	Způsob hodnocení - následující škálou 1 až 4:
2. Splnění zadání	1=zadání splněno, 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno
Popis kritéria: Posuďte, zda předložená ZP splňuje zadání. V komentáři uveďte body zadání, které nebyly zcela splněny, případně rozšíření ZP oproti původnímu zadání. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.	
Komentář: Kompilátor byl studentem vytvořen a podporuje dostatečnou podmnožinu jazyka. Výsledná implementace byla co do funkčnosti otestována a bylo provedeno i porovnání rychlosti výsledné spustitelné verze jednoho zdrojového programu s dalšími dvěma interpretry.	
Hodnotící kritérium:	Způsob hodnocení - následující škálou 1 až 4:
3. Rozsah písemné zprávy	1=splňuje požadavky, 2=splňuje požadavky s menšími výhradami, 3=splňuje požadavky s většími výhradami, 4=nesplňuje požadavky
Popis kritéria: Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části.	
Komentář: Kapitoly Analysis, Design a Implementation jsou velmi rozsáhlé a podrobné. Bohužel kapitola Testing je dle mého názoru velmi krátká na to, jak jednoduše se dá naplnit užitečnými výsledky měření. Výsledné kompilované verze programů mohou mít při běhu odlišné vlastnosti oproti běhu v interpretu. Například program může být náročný na práci s pamětí nebo ne, může využívat seznamů nebo převážně jen celých čísel a podobně.	
Hodnotící kritérium:	Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):
4. Věcná a logická úroveň práce	80 (B)
Popis kritéria: Posuďte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti. Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře.	

Komentář:

Některé technické návrhy řešení z kapitoly 1 Analýza bych spíše viděl v kapitole 2 Design (poslední odstavec ze sekce Type system nebo seznam řešení a zbytek sekce Dynamic calls), přičemž stejně jsou dále v podstatě zopakovány.

Dále bych z kapitoly 2 Design také posunul implementační detaily do kapitoly implementace (například poslední odstavec sekce Lexical analysis). V kapitole Design v sekci Functions and closures mi přijde zvláštní, že by nebylo známo, jestli je volaná funkce variadická nebo ne. Kdy jindy by to mělo být známo? Během kompilace ta informace známá je, proč není tedy známá v runtime? Je toto vlastnost jazyka Scheme nebo šlo o Vaše rozhodnutí? Optimalizace shrnuté v kapitole design jsou dle mého názoru zbytečné a navíc jsou popsány naprosto bez přidané hodnoty (např. Dead Instruction Elimination), přičemž jediná a dále více popsaná je Tail Call Elimination.

V kapitole Testing sekci Performance není jasná metodika měření. Kolikrát bylo měření opakováno, zda je čas součtem nebo průměrem opakování a podobně. Pro objektivitu měření chybí informace o použitém HW a verzích porovnávaných interpretrů jazyka Scheme. Násobení matic je také jen jeden program z testovací sady použité pro ověření funkčnosti. Proč nebyl podobným způsobem využit nějaký další škálovatelný program například SAT?

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

5. Formální úroveň práce

95 (A)

Popis kritéria:

Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 12/2014, článek 3.

Komentář:

Práce obsahuje dle mého názoru zbytečné dopředné odkazy. Kód (c++) -- konkrétně elipsy -- jsou sázeny s nekonzistentně velkými mezerami. Raději bych viděl text práce psaný v trpném rodě než v plurálu, nicméně to není chyba. Občas je text nepřirozeně přerušen velkým obrázkem a pokračuje až další sekci na další straně. V blocích listing občas přetéká text pravou hranu textu.

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

6. Práce se zdroji

100 (A)

Popis kritéria:

Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení ZP. Charakterizujte výběr studijních pramenů. Posuďte, zda student využil všechny relevantní zdroje nebo zda se pokoušel řešit již vyřešené problémy. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků a úvah, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami.

Komentář:

V kapitole Implementation details je zmíněna třída GetRange, která je dle mého názoru zbytečná, když standardní knihovna definuje od verze c++14 typ index_sequence, který má stejné chování. Toto není ani tak výtka jako spíše poznámka. Vlastní citace jsou v pořádku.

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

7. Hodnocení výsledků, publikační výstupy a ocenění

80 (B)

Popis kritéria:

Vyjádřete se k úrovni dosažených hlavních výsledků ZP, např. k úrovni teoretických výsledků, nebo k úrovni a funkčnosti technického nebo programového vytvořeného řešení, apod. Případně také zhodnoťte, zda software nebo zdrojové texty, které nevytvořil sám student, byly v ZP použity v souladu s licenčními podmínkami a autorským právem. Popište případnou publikační činnost a získaná ocenění související s řešením této ZP.

Komentář:

Existují pokusy o frontend pro programovací jazyk Scheme pro LLVM (například Scheme R7RS frontend for LLVM). Většina není dále vyvíjena. Pokud se tento frontend vyvíjený v rámci diplomové práce udrží ve vývoji, je to jistě výborně. Samotné vypracování je pro studenta vzhledem k dalšímu pokračování studia v týmu ELE na doktorském studiu přínosné. Ukázkové programy i další jednoduché programy zvládne frontend zkompilovat. V případě složitějších jsem narazil na neužitečné informace o syntaktických chybách -- řádek ani sloupec chyby není zobrazen. V případě dalších programů jsem dokonce dostal segmentation fault.

Hodnotící kritérium:

Způsob hodnocení - nehodnotí se

8. Komentář o využitelnosti výsledků

Popis kritéria:

Uveďte, zda hlavní výsledky ZP rozšiřují již publikované známé výsledky a/nebo přinášející zcela nové poznatky. Uveďte možnosti využití výsledků ZP v praxi.

Komentář:

Viz předchozí bod.

Hodnotící kritérium:

Způsob hodnocení - nehodnotí se

9. Otázky k obhajobě

Popis kritéria:

Uveďte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).

Otázky:

Jaká byla metodika měření a jaký je výkon llscheme v porovnání s, v práci zmíněnými, interpretry na programu SAT?

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

10. Celkové hodnocení

87 (B)

Popis kritéria:

Shrňte stránky ZP studenta, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení **nemusí** být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích 1 až 9.

Text hodnocení:

Výsledek práce je, co se týče programátorské práce, dobrý. Uživatelská přívětivost je na horší úrovni. Kontrola vstupu je sice syntaxí řízeným překladem provedena, ale dále program může pro chybné vstupy spadnout na neoprávněný přístup do paměti. Pro jednodušší programy z podporované podmnožiny je frontend funkční. Nejvíce mě ale v hodnocení ovlivnilo nedostatečné měření. Jedná se o prakticky zaměřenou práci, kterou navíc lze velmi jednoduše porovnat s dalšími interpretry. Celkově hodnotím práci 87 body stupněm B (velmi dobře).

Podpis oponenta práce: