Posudek oponenta diplomové práce

Autor/autorka práce: **František Šimeček**

Název práce: **Automatický převod programů napsaných v jazyce Java do dalších jazyků (C/C++/FreePascal)**

Obsah práce

Cílem práce je tvorba transpileru, který bude překládat programy napsané v definované podmnožině jazyka Java do C, C++ a FreePascalu. Práce má dvě zásadní části, jedna je zpracování podmnožiny jazyka a jedna je převod části běžných knihoven Javy do jiných prostředí nebo hledání jejich vhodných alternativ.

Samotná práce se v první části věnuje obecně problematice překladačů a transpilerů, následuje rozbor vlastností jazyka Java a jeho paralel s cílovými jazyku a konečně implementace a testy programu. Text práce je dobře strukturovaný a dobře čitelný, ne se všemi tvrzeními v něm bych ale souhlasil.

Především popis regulárních výrazů a gramatik (na s. 13) je značně neformální. V dalším textu jsou ale používány pojmy, které z formálního popisu vychází a očekávají jeho znalost. Dále mě překvapuje volba javascriptu jako zástupce mezijazka, očekával bych spíš bytecode, CIL, LLVM IR nebo něco podobného. Také nemohu souhlasit s tím, že mezijazyk nemusí být umělý a nemusí mít pevně danou strukturu – naopak bych tvrdil, že je vždy umělý a že má vždy pevně danou strukturu, byť nemusí být explicitně popsána formální gramatikou (což je třeba případ této DP).

Dále příliš nesouhlasím s tvrzeními v kapitole 5.3 týkajících se tabulky symbolů. Řekl bych, že tabulka symbolů v tomto případě nemusí být závislá na vlastnostech cílového jazyka. Očekával bych, že ji parser Javy vytvoří, nezávisle na tom do jakého jazyka se bude překládat, a až příslušný modul zodpovědný za generování kódu pro cílový jazyk ji zpracuje podle potřeby. Konečně, v kapitole 5.4.2 je řeč o lambda výrazech a složitosti jejich překladu, nikde ale není zmíněno, jestli program umí nebo neumí zpracovat anonymní vnitřní třídy, na které lze lambda výrazy vždy bez problémů převést.

Kvalita řešení a dosažených výsledků

Vytvořený program je použitelný a funkční, velmi důkladně otestovaný a schopný zpracovat všechny úlohy na Pilsprog z minulých let a to považuji za velmi dobrý výkon. Byť by program mohl být lépe strukturován, zejména z hlediska oddělení parsování, reprezentace programu a generování cílového jazyka, je jeho zdrojový text čitelný a srozumitelný. Technika implementace – rekurzivní sestup – je rozhodně vhodná. Očekával bych ale, že rekurzivní sestup bude založen spíš na gramatice zdrojového jazyka spíš než na jeho intuitivním pochopení. To je zejména vidět na analýze výrazů, pro jejichž překlad je místo rekurzivního sestupu použita varianta zásobníkového automatu. Na druhé straně tato implementace je plně funkční a dokáže si poradit s netriviálními algoritmy z PilsProgu.

Formální úroveň

Text práce je dobře čitelný a srozumitelný, chybí v něm ale řada teoretických věcí týkajících se překladu jazyka, které bych v něm očekával. Zdrojové texty práce jsou v pořádku, program je přehledný a alespoň stručně komentovaný. Textu by prospěla podrobnější kontrola, na několika místech se vyskytují zvláštní výrazy (např. zparsovat na s. 12).

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
katedra informatiky a výpočetní techniky

Práce s literaturou

Práce se odkazuje na 34 zdrojů, které jsou v textu použity. Bohužel ne vždy je autor skutečně využil – odkazuje např. na popis tvorby rekurzivního sestupu na základě gramatiky jazyka. Odkaz 9 (Compiler construction) je zřejmě špatně zformátován.

Splnění zadání

Všechny body zadání považuji za splněné.

Přes uvedené výhrady navrhuji hodnocení známkou **velmi dobře**. Důslednější využití formálních metod by autorovi pravděpodobně usnadnilo práci, ale i cesta, kterou zvolil, vedla k efektivnímu nástroji použitelnému pro převod stávajících programu. Práci doporučuji k obhajobě.

V Plzni 1.6.2017

Ing. Richard Lipka, Ph.D.

SOUHLASÍ
S ORIGINÁLEM
Západočeská univerzita v Plzni
Fakulta aplikovaných věd
katedra informatiky a výpočetní techniky
①