



Posudek oponenta závěrečné práce

Student: Lukáš Simulík
Oponent práce: Ing. Tomáš Pecka
Název práce: Transpilátor z jazyka PHP do jazyka Go
Obor: Teoretická informatika

Datum vytvoření: 15. 6. 2020

<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – následující škálou 1 až 4:</i>
1. Splnění zadání	<u>1=zadání splněno,</u> 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno
<i>Popis kritéria:</i> Posuďte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posuďte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.	
<i>Komentář:</i> Práce se zabývá překladem kódů ze zdrojového jazyka PHP do jazyka Go za účelem zrychlení vykonávání skriptů. Téma je to určitě náročnější, přestože se měl autor omezit jen na základní konstrukce. Zadání bylo splněno.	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):</i>
2. Písemná část práce	60 (D)
<i>Popis kritéria:</i> Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále 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. Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3. Posuďte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišený od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.	

Komentář:

Informační bohatost jednotlivých kapitol kolísá. První tři kapitoly (úvod do zrychlování interpretace obecně i PHP) jsou přínosné. Chybí nějaký přehled rozdílů mezi PHP a Go, ty se čtenář dozví až průběžně v kapitole 5 (Realizace). Kapitola 4 se má (podle autora) zabývat částmi kompilátoru zajímavými pro jeho práci - tedy lexikální a syntaktickou analýzou a generováním kódu. O generování kódu zde nic není. Autor zhruba popisuje LL parsing, ale použitá parsovací knihovna [34] generuje LR parser. Vůbec se nevěnuje vnitřním reprezentacím kódu.

Z textu není úplně jasné jak proces překladu probíhá. Jaké operace a datové struktury musel autor doimplementovat a jaké už byly poskytnuty z [34] (kdo vytváří AST?) a co se dále s AST děje (provádí se transformace?).

U přístupu k překladům jednotlivých problémových konstrukcí by byly vhodné příklady. Chybí také nějaký soupis podporované podmnožiny vstupního jazyka.

Zvláště v druhé polovině textu je větší množství překlepů a nesmyslných vět. Citované zdroje jsou v pořádku. Z typografického hlediska je práce celkem zdařilá, našel jsem jen malý počet problémů.

Další poznámky:

- V úvodu byl slib, že dojde k pokusům o překlad již existujících projektů, ale v textu jsem o tom dále nic nenašel, kromě zmínky v závěru, že se kvůli tomu přidávala podpora transformace kódu pro připojení k (My)SQL databázím.
- Přetékající anglický abstrakt nepůsobí dobře.
- Sekce 4.1: Koncové stavy konečného automatu jsou soustavně označovány jako "konečné" nebo "ukončovací". Konečný automat se skládá z pěti částí, ale vyjmenované jsou jen čtyři. Dále v textu se pak konečně vyskytne ta pátá - překladová funkce, zde ale asi spíše přechodová. Nechápu poznámku, proč by obecný automat měl mít jen jeden koncový stav.
- Sekce 5.4: První odstavec je zmatený. Není jasné, zda se mluví o proměnných, funkcích, nebo obojím a co je vlastně ten problém, o kterém se mluví.
- Sekce 5.10 má popisovat "Funkce a proměnný počet argumentů", popisuje ale jen překlad funkcí s výchozími hodnotami argumentů (Default argument values), namísto očekávaného libovolného počtu argumentů (Variable-length argument lists), nebo obojího.
- Sirotek na str. 22, nezvyklý způsob zápisu při více citovaných zdrojích ([x], [y]). Výpisy kódů není vhodné značit jako obrázek. Student se často odkazuje jen číslem (např. v 2.1, v 5.3) a není na první pohled jasné, zda míří na sekci v textu, či obrázek, nebo se odkazuje na obrázek jako na příklad (5.1).

Hodnotící kritérium:

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

3. Nepísemná část, přílohy

70 (C)

Popis kritéria:

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Významná a experimentální práce – opakovatelnost experimentů

Komentář:

Jako proof-of-concept překladu se, na příkladech dodaných autorem, výsledná aplikace osvědčila. Přeložený kód vypadá podobně jako kód na vstupu. Pro ukázkové vstupy aplikace (až výjimky, viz níže) funguje. S jinými vstupy je to horší, neboť spousta konstrukcí není ještě v aktuální implementaci podporována.

K aplikaci mám však pár zásadních připomínek: Chybí dokumentace či manuál, kód je téměř neokomentovaný. Orientace v projektu je obtížná a některé zdrojové soubory jsou velmi dlouhé. Často jsem musel postupovat přístupem pokus/omyl co vlastně překladač umí. Při překladu nepodporované konstrukce (např. 'echo \$\$var' nebo třídy) se program ukončí s chybou, která ne vždy jasně specifikuje co se stalo, a ani neoznámí, kde k chybě došlo. Autor v sekci 5.2 píše, že informace o umístění prvků v souboru ignoruje, ale přesně pro tyto případy jsou právě užitečné.

Očekával bych větší množinu testů, na podchycení chování překladače je jich dle mého názoru málo.

Další poznámky:

- Hned při prvním pokusu se překlad souboru examples/01.php povedl, ale spuštění výsledku ne, kvůli (asi) chybějícímu argumentu u funkce Fprintf. Po přečtení dokumentace této funkce jsem problém vyřešil úpravou výsledku - přidáním argumentu konverzního řetězce. Při pokusu o překlad bez serverové části se program spustil a proběhl v pořádku.
- Přeložený program nefunguje pokud se snažím přeložit skript nazvaný "foo.php", který přes require/include klauzuli vkládá stejně nazvaný skript v jiném adresáři ('subdir/foo.php'), vytvoří jen jeden soubor a v něm je pouze obsah vkládaného skriptu ('subdir/foo.php').
- Při pokusu o výpis klíčů a hodnot vytvořeného asociativního pole (string->string) přes foreach se v přeloženém kódu nevypíše klíč. Kód v std/strings.go naznačuje problém v funkcích Concat a interfaceToString, protože klíč je neočekávaného typu, na což se reaguje výpisem prázdného řetězce.
- Std balík není dostatečně modulární a tak všechny programy, které ho importují vyžadují instalaci mysql driverů, přestože se připojení nepoužívá.

Hodnotící kritérium:

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

4. Hodnocení výsledků, jejich využitelnost

75 (C)

Popis kritéria:

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

Komentář:

Aplikace může sloužit jako proof-of-concept překladu skriptů z PHP do Go. K tomu, aby výsledná aplikace vypadala rozumně a dala se veřejně publikovat (a například nechala rozšiřovat komunitou) bude ještě potřeba kód "učesat" a doplnit, například o překlad tříd (v dnešní době podstatná feature), či autorem navrhované komentáře upřesňující typy proměnných.

Hodnotící kritérium:

Způsob hodnocení – nehodnotí se

5. Otázky k obhajobě

Popis kritéria:

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

Otázky:

- Můžete prosím lépe objasnit, které části překladače jsou Vaše práce a které za vás řeší použité skripty třetích stran? Hlavně v souvislosti s tvorbou AST.

- Jaké důvody Vás vedly k ignorování informace o umístění prvků ve vstupním souboru (sekce 5.2)?

Hodnotící kritérium:

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

6. Celkové hodnocení

70 (C)

Popis kritéria:

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.

Text hodnocení:

Bohužel, písemná část vykazuje určité nedostatky. Implementační část funguje a výsledná aplikace umí překládat základní konstrukce z jazyka PHP do jazyka Go. Práce má jistě potenciál, ale výslednému softwarovému dílu se bude chtít ještě nějakou dobu věnovat a to nejen kvůli ještě nepodporovaným konstrukcím. Na druhou stranu, student si musel poradit se spoustou netriviálních problémů, které přináší převod programů z PHP do Go.

Práci doporučuji k obhajobě a navrhuji 70 bodů, tedy ještě známku C.

Podpis oponenta práce: