

Posudek oponenta bakalářské práce

Student: Sabol Matúš

Téma: Gramatická formalizace překladu a její implementace (id 19042)

Oponent: Krčmář Radim, Ing., UIFS FIT VUT

- 1. Náročnost zadání** **obtížnější zadání**
Překladové gramatiky a publikace nutné k jejich pochopení jsou na úrovni doktorského studia.
- 2. Splnění požadavků zadání** **zadání splněno**

- 3. Rozsah technické zprávy** **splňuje pouze minimální požadavky**
Práce je na spodní hranici obvyklého rozsahu, když přihlédneme k vyšší vyjadřovací síle formálního zápisu.

- 4. Prezenční úroveň předložené práce** **55 b. (E)**
Nedostatky:

- Práce je sice rozdělena na teoretickou a implementační část, ale přesto se v implementační části vyskytují důležité definice mimochodem po jejich prvním použití, e.g. LL gramatiky.
- Důvod konstrukce zásobníkového převodníku z překladové gramatiky není srozumitelný.
- Převzaté definice nejsou vždy úplné (e.g. chybí derivace u překladových gramatik) a jednoduchým příkladem je naopak věnováno zbytečně mnoho prostoru (e.g. opakování vstupní a výstupní gramatiky bez zřejmého záměru).

- 5. Formální úprava technické zprávy** **68 b. (D)**
Práce je psána srozumitelnou angličtinou s občasnými překlepy. Mezi typografické prohřešky patří český standard uvozovek a hlavně umístění interpunkce na začátek řádku za matematickým výrazem.

- 6. Práce s literaturou** **59 b. (E)**
Zdroje jsou voleny vhodně, ale práce s nimi je slabší:

- Po teoretické kapitole už se v textu necituje, ačkoliv některé tvrzení musejí pocházet z literatury, e.g. tvrzení okolo "recursive descent".
- Do převzaté definice překladové gramatiky je zavedena formální chyba. (Formulace "nonterminals in beta are a permutation of the non-terminals in alpha.")
- Práce přichází s vlastní překladovou gramatikou pro jednoduché výrazy, ačkoliv v literatuře lze najít hotovou a vhodnější.

- 7. Realizační výstup** **49 b. (F)**
Výstupem jsou dvě části:

1. Algoritmus zpracování překladových gramatik. Zvolená překladová gramatika pro převod notace nevyjadřuje prioritu operátorů a tento nedostatek je obcházen úzkou specializací překladového algoritmu, který předzpracovává operátory ze vstupního řetězce podle předem známé priority.
2. Implementace algoritmu v jazyce C++. Jedná se o minimalistický konzolový program (~440 řadků), který převádí infixovou notaci na postfixovou. Dokumentace je uspokojivá ale program selhává se vstupy obsahujícími více podvýrazů v závorkách, e.g. $(a+b)^*(b+c)$, a cyklí nad některými nevalidními vstupy, e.g. `a++a`.

Na bakalářské úrovni je méně obecné řešení první části pochopitelné, ale nefunkční implementaci jednoduchého převodu hodnotím pouze jako téměř dostatečnou.

- 8. Využitelnost výsledků**
Na převod notací existují lepší programy a algoritmus konstrukce je použitelný pouze na velmi malou množinu gramatik.

9. Otázky k obhajobě

- Proč je zásobníkový převodník kombinován s překladovou gramatikou?
- Jakým způsobem lze vyjádřit prioritu operátorů v bezkontextové překladové gramatice?

- Kde jsou chyby kvůli kterým program nepřekládá validní vstup a jinde cyklí?

10. Souhrnné hodnocení

60 b. uspokojivě (D)

Práce je podprůměrná ve všech ohledech a chybná implementace vzbuzuje pochyby o vhodnosti zvoleného přístupu. Vzhledem k vyšší náročnosti zadání navrhuji, v případě dobrého vysvětlení chyby, mírnější hodnocení na spodní hranici D.

Prohlášení: Uděluji VUT v Brně souhlas ke zveřejnění tohoto posudku v listinné i elektronické formě.

V Brně dne: 6. června 2017

.....
podpis