

Posudek na diplomovou práci

Peter Kuziel

Konstruktory přívětivých analyzátorů pro Javu

Tato práce navazuje na dlouholetý výzkum vedoucího práce na poli tzv. přívětivých syntaktických analyzátorů. Jde o zcela novou implementaci konstruktora syntaktického analyzátoru (v jazyce Java a pro jazyk Java), na druhé straně práce neobsahuje žádný pokrok v teoretické oblasti, pouze opakuje výsledky a implementuje postupy vytvořené v předchozích pracích jiných autorů. Vzhledem k tomu, že jde o konstruktor, práce nezahrnuje to nejzajímavější z dosud publikované teorie přívětivých gramatik – rozšiřitelnost za běhu. Obsahem práce je poměrně jednoduchá základní metoda přívětivé analýzy pro $k=1$; nejvýznamnějším řešeným problémem je tedy zapojení semantických akcí, které autor implementoval ve třech variantách – jednoduchý generator výstupu, akce zapsané v pseudokódu (navrženém v předešlých pracech na toto téma) a akce zapsané v jazyce Java.

Text práce začíná (po nezbytném úvodu) pokusem o kategorizaci metod syntaktické analýzy z hlediska praktického používání parserů a jejich generátorů. Autorův rozhled je ovšem omezen na LL(k) a LALR(1) metody a zanedbává významné alternativy jako GLR. Klasifikace se zcela vyhýbá jakýmkoliv formálnějším pojmům jako derivace, atributová gramatika, syntaxí řízený překlad, rekurzivní sestup apod., které by v souvislosti se schopnostmi praktických analyzátorů měly být zmíněny.

Následuje 26-stránkový popis metody přívětivé analýzy a souvisejících problémů. Vzhledem k tomu, že přínos autora je v této části pravděpodobně pouze v překladu do slovenštiny, je na místě otázka, zda nejde o zbytečné plýtvání časem autora (i čtenáře), které by bylo možno nahradit odkazy (které jsou ostatně u každého odstavce poctivě uvedeny). Řada zmíněných pojmů je navíc definována zcela zbytečně, neboť dále není využívána.

Zbytek textu popisuje především formát souboru s lexikální a gramatickou strukturou jazyka, včetně sémantické části. Struktura generovaného zdrojového textu je uvedena pouze velmi nepřehledným příkladem a technicky zajímavé detaily, kterých by se jistě našla řada, popsány nejsou. Zcela chybí zdůvodnění zvoleného řešení či diskuse alternativ, zejména u rozhodnutí, která ovlivňují vnější rozhraní systému.

V textu chybí kvantitativní hodnocení předloženého softwaru, přitom jde o jednu z mála vlastností produktu, za které se autor nemusí stydět – čas konstrukce parseru je zanedbatelný a rychlost generovaného parseru je lepší než 10 000 řádek za sekundu.

Praktická použitelnost díla je degradována řadou nedomyšleností a nedodělků: Generovaný parser vždy čte soubor jménem "input_" a o nějaké možnosti přesměrování na nesouborový vstup nelze uvažovat vzhledem k absenci vhodného rozhraní. Chybová diagnostika jak konstruktora, tak generovaného parseru je nulová. Generované třídy se vždy jmenují stejně, takže koexistence dvou parserů v jednom programu není možná. Lze tedy konstatovat, že software je funkční na úrovni proof-of-concept, to je ovšem vzhledem k existenci několika předchozích implementací téhož algoritmu poněkud málo.

Z výše uvedených připomínek vyplývá, že implementace není dotažena do úrovně, kterou by bylo možné publikovat jako prakticky použitelný nástroj. Vzhledem k tomu, že jiný než

implementační obsah práce nemá, jde o závažný nedostatek, který spolu s uvedenými nedostatky textu vyvolává pochybnosti o obhajitelnosti práce. Protože však jde o dílo z netriviální oblasti a základní cíle práce byly víceméně naplněny, doporučuji tuto práci připustit k obhajobě.

16.1.2009



David Bednárek