

Identifikace limitací dosavadních technologií v kontextu projektu

Z pohledu řešení VUT a jeho dosavadní úlohy v řešení projektu VLAM lze nalézt tyto existující technologie vhodné pro analýzu formálních jazyků a z nich vyplývající omezení:

Nástroje pro převod bezkontextových gramatik na nástroje syntaktické analýzy (výběr):

SLK – praktický nástroj používaný pro generování LL(k) tabulky a analyzátoru založeného na prediktivní analýze podporované vytvořenou LL(k) tabulkou (<http://home.earthlink.net/~slkpg>). Nástroj je vhodný pro náročnější gramatiky (jako např. gramatika VHDL), protože je schopen vytvářet optimalizované LL(k) tabulky pro i poměrně veliké konstantní k (až desítky).

Bison – další nástroj na generování analyzátorů opět založený na tabulce, ale pracující metodou zdola nahoru a s vyšší vyjadřovací silou (LALR, avšak schopen v případě využití obecného ale méně efektivního překladu zpracování třídy LR jazyků). Všechna pravidla jsou zapisována ve speciálním formátu zavedeném již mnohem starším nástrojem YACC (Yet Another Compiler Compiler). Umožňuje také každé pravidlo obohatit sémantickou akcí (např. volání lib. funkce jazyka C) a tím získat dostatečně detailní popis jazyka i jeho zpracování, aby mohl být vytvořen kompletní překladač studovaného jazyka.

<http://www.gnu.org/software/bison/bison.html>

<http://gnuwin32.sourceforge.net/packages/bison.htm>

Nástroje pro převod regulárních gramatik na nástroje lexikální analýzy (výběr):

Flex – nástroj sloužící pro předzpracování vstupního souboru před samotným překladem. Program rozdělí zpracovávanou větu (celý soubor) na základní jazykové entity nazývané lexémy (navíc obohacené dodatečnými atributy, pak získáme tzv. tokeny). Tyto entity jsou dále předloženy syntaktickému analyzátoru ke zpracování. Analyzátor může být například generován právě nástroji jako Bison nebo SLK.

<http://www.gnu.org/software/flex/flex.html>

<http://gnuwin32.sourceforge.net/packages/flex.htm>

Další nástroje vhodné pro zpracování (výběr):

Dokumentační jazyky – metodologie, jak prostřednictvím rozšířených komentářů uvnitř samotného zdrojového kódu v různých programovacích jazycích uchovávat informace potřebné pro automatické avšak původně netypické analyzování zdrojových kódů. Nejčastěji je účelem možnost automatické tvorby hypertextové dokumentace a její pohodlná úprava na stejném místě, kde probíhá samotná změna zdrojového kódu (a kde je tudíž i umístěn odpovědný komentář). Další možností je uchování doplňujících informací pro překladač či jiný druh automatické analýzy textu (tzv. značky, angl. tags). Příkladem může být JavaDoc v IDE NetBeans, kde pomocí speciálních značek v komentářích označují automaticky generovaný kód, který nemá programátor povolen editovat.

Např. VHDLDOC: <http://schwick.web.cern.ch/schwick/vhdl/doc/>

Šířeji podané informace lze nalézt na

<http://engtech.wordpress.com/2006/07/05/inline-source-code-documentation-language-independent/>

GCC – GNU Compiler Collection – open source sada nástrojů (včetně toolchain) pro zpracování a překlad různých programovacích jazyků (především C, C++, Java, Pascal, Ada, aj.)

<http://gcc.gnu.org/>

Chybějící vstupy

Kromě nástrojů byla také analyzována možnost využití existujících formálních popisů jazyka VHDL. Všechny dostupné zdroje však neobsahovali specifikaci jazyka podle současné normy VHDL 2002.

V rámci řešení dílčího cíle 1 tak byly vyvinuty 2 gramatiky, které se snaží různým postupem řešit komplexnost jazyka a jeho značné kontextové vazby.

Tyto gramatiky budou uplatněny v dalších krocích řešení.

Shrnutí

I když by se mohlo zdát, že daná množina nástrojů je dostatečná, tak pro náročný úkol analýzy VHDL v celé jeho šíři představuje problém, kdy neexistuje vhodný nástroj umožňující popsat a definovat kontextové vazby a z nich vyplývající důsledky – identifikace klíčových míst pro důkladnější analýzu.

Ani nástroje pro důkladnou analýzu již vytipovaných a předchystaných míst neexistují, nebo minimálně v době psaní tohoto textu je nebylo možné volně využít.

Jelikož tedy chybí nástroj pro zpracování kontextových jazyků (podobně jako Bison pro jazyky bezkontextové), bude jej třeba vyvinout v případě, že analýza bezkontextového typu, založená na již vytvořených gramatikách a obohacená o dodatečnou sémantickou analýzu, nebude dostačovat. K tomu disponuje řešitelský tým VUT dostatečnými znalostmi i výzkumným potenciálem:

Kolář, D., Meduna, A.: Regulated Automata: From Theory towards Applications, In: Proceeding of 8th International Conference on Information Systems Implementation and Modelling ISIM'05, Ostrava, CZ, MARQ, 2005, pp. 33-48, ISBN 80-86840-09-3.

Kolář, D.: Pushdown Automata: Another Extensions and Transformations, Brno, CZ, FIT VUT, 2005, p. 76.

Meduna, A., Kolář, D.: One-Turn Regulated Pushdown Automata and Their Reduction, In: Fundamenta Informaticae, roč. 2002, č. 16, Amsterdam, NL, pp. 399-405, ISSN 0169-2968.

Kolář, D., Meduna, A.: Regulated Pushdown Automata, In: Acta Cybernetica, roč. 2000, č. 4, US, pp. 653-664, ISSN 0324-721X.

Výzkum v této oblasti bude probíhat paralelně a nezávisle na hlavním směru, který se bude snažit využít a obohatit již existující a v rámci projektu vytvořené bezkontextové gramatiky.

Vytvořil:
Z. Křivka
D. Kolář