

THE CONVERTER LANGUAGE WATCOM FORTRAN 77 TO LANGUAGE C++

Martin ŠTĚPÁNEK, Master Degree Programme (5)
Dept. of Information Systems, FIT, BUT
E-mail: xstepa20@stud.fit.vutbr.cz

Supervised by: Dr. Dušan Kolář

ABSTRACT

Watcom Fortran 77 is programming language which is used for exact technical calculations and is connected with well sophisticated input / output operations. The syntax of this language is not too much different from the syntax of language C or C++. The goal of this project is the creation of the converter with the same behaviour after the compilation as in language Watcom Fortran 77. The converter uses the library that was implemented in the year-class project.

The converter goes through source code and generates the resultant source code in C++. The converter transforms constructs: the definition and initialization variables, basic loops in the language Watcom Fortran 77 and also conditional and unconditional jumps. The converter supposes that the source code of Watcom Fortran 77 is syntactically and semantically correct and therefore the converter does not do syntactic and semantic control.

1 ÚVOD

Cílem tohoto projektu je vytvořit převodník z jazyka Watcom Fortran 77 do jazyka C++ tak, aby se po převodu výsledný kód choval stejně jako ve Watcom Fortran 77.

Převodník ke své činnosti využívá lexikální analyzátor, který provádí načtení a identifikování jednotlivých lexémů. Tyto lexémy jsou využívány syntaktickým analyzátozem, který podle typu lexému vybere příslušné gramatické pravidlo z rozkladové tabulky. Podle daného pravidla provádí příslušný překlad jednotlivých konstrukcí.

Převodník také využívá „runtimeových“ knihoven (knihovny se využívají až při kompilaci C++ kompilátorem), které pokrývají V/V operace ve Watcom Fortran 77. Tyto knihovny byly vytvořeny v ročníkovém projektu (viz [2]). Dále využívá vytvořenou matematickou knihovnu, kde jsou implementovány některé matematické funkce, které jazyk C++ ani C nenabízí, např. nalezení maxima či minima z n čísel.

2 ROZBOR

Cílem této kapitoly je seznámit se s použitými analyzátory a hlavně ukázat na odlišnosti jazyku Watcom Fortran 77 od jazyka C++.

2.1 LEXIKÁLNÍ ANALYZÁTOR

Lexikální analyzátor čte jednotlivé znaky ze vstupu a provádí jejich správnou identifikaci. Jazyk Watcom Fortran 77 dovoluje používat klíčová slova také jako identifikátor proměnné. Musel jsem tedy podle dalších lexémů správně vyhodnotit, zda se v daný moment jedná o klíčové slovo nebo o identifikátor proměnné.

2.2 SYNTAKTICKÝ ANALYZÁTOR

Po správném identifikování typu lexému se podle něj vybere gramatické pravidlo z rozkladové tabulky. Syntaktický analyzátor potom syntaktickou analýzou shora dolů prochází vstupní zdrojový soubor a provádí generování příslušného výstupního kódu v jazyce C++.

Protože se jedná o převod z Watcom Fortranu 77 do jazyka C++, měly by být zdrojové soubory již syntakticky správně a tedy možné přeložit kompilátorem Watcom Fortran 77. Můj převodník neprovádí žádné zotavení z chyb a také neprovádí sémantickou kontrolu, protože toto by již mělo být ošetřeno správností vstupních dat. Pokud by se ovšem nějaká chyba našla, převodník vytvoří chybový soubor, kde jsou vypsána jednotlivá gramatická pravidla a jako poslední v tomto souboru bude uvedeno chybné pravidlo.

2.3 GRAMATIKA PŘEVODNÍKU

Pro gramatický popis jazyka Watcom Fortran 77 jsem použil LL(1) gramatiku. Gramatika LL(1) prochází text zleva doprava, provádí levý rozklad a ke správnému výběru gramatického pravidla z rozkladové tabulky stačí jeden lexém. Ovšem u jazyka Watcom Fortran 77 v některých případech nešlo podle prvního lexému správně určit gramatické pravidlo. Proto jsem musel podle dalších symbolů správně určit, o které pravidlo se v daný moment jedná, vybrat ho a dále opět provádět syntaktický překlad.

2.4 GENEROVÁNÍ VÝSTUPNÍHO KÓDU

Kód se generuje přímo při syntaktickém průchodu zdrojových dat. U velkých a složitých projektů se musí použít dva průchody. To je způsobeno další odlišností jazyka Watcom Fortran 77. Ten totiž umožňuje definovat externí funkce jako proměnné. Při prvním průchodu se zjistí všechny externí funkce a uloží se do tabulky symbolů. V druhém průchodu již převodník ví, jaké externí funkce jsou použity. Tedy jakmile narazí opět na deklaraci externí funkce, již ví, že to není proměnná, ale externí funkce, jejíž definice je v jiném modulu (souboru).

Kód se ukládá do dynamického seznamu, kdy každý řádek v tomto seznamu je jeden příkaz ve Watcom Fortranu. Další zvláštností Fortranu je definování datových typů formálních parametrů funkce až uvnitř této funkce. Jazyk C++ vyžaduje datové typy hned při definování těchto parametrů. Dynamický seznam mi umožňuje vrátit se již ke starým příkazům a tedy v tomto případě dopsání datových typů k jednotlivým parametrům funkce.

3 VYTVOŘENÉ A POUŽITÉ KNIHOVNY

Tyto knihovny byly vytvořeny v rámci ročníkového projektu (viz [2]) a zajišťují stejné programování V/V operací jako ve Watcom Fortran 77. Tyto knihovny se využívají až při kompilaci C++ překladačem.

3.1 PRÁCE SE SOUBORY

Watcom Fortran 77 přistupuje k souborům pomocí celočíselného ukazatele, kdežto jazyk C++ pomocí symbolického jména. Proto jsem musel vytvořit třídu `__cMyPool`, která obsahuje metodu `GetUnit (int cislo)`. Tato metoda vrací ukazatel na třídu `__cMyFile`, která pomocí parametru `cislo` zpřístupní daný soubor. Pokud ještě takový soubor nebyl vytvořen, vytvoří se nová třída, která je prázdná a neobsahuje tedy žádná data.

Uložení informací do daného objektu se provádí pomocí metod, které odpovídají jednotlivým specifikátorům. Tyto specifikátory v jazyku Watcom Fortran 77 nastavují a mění vlastnosti jednotlivých souborů, např. přístup k souborů nebo otevření souborů.

Pro práci se soubory jsou v jazyce Watcom Fortran 77 použity příkazy `OPEN`, `CLOSE`, `INQUIRE`, `BACKSPACE` a `REWIND`.

3.2 ČTENÍ A ZÁPIS

Ke čtení i zápisu v jazyce Watcom Fortran 77 se může použít stejný formátovací řetězec. Tento formátovací řetězec jsem musel nejprve analyzovat a rozdělit jej na jednotlivé složky. Každá složka formátu je opět nová třída, která obsahuje veškeré informace o dané složce, např. délku, obsah, typ atd.

Pro zápis resp. čtení pomocí formátovacího řetězce jsem opět vytvořil metody `Write` resp. `Read`. Tyto metody dostávají jako vstupní parametry formátovací řetězec a seznam proměnných ze/do kterých se má zapisovat/číst. Další zvláštností Fortranu je zápis resp. čtení do proměnné datového typu `CHARACTER`. Pro tento zápis/čtení jsem vytvořil metody `WriteProm/ReadProm`, které mají stejné vstupní parametry jako metody předešlé a navíc proměnou, do které se má zapisovat/číst.

Pokud se formátovací řetězec nevyužívá, vytvořil jsem metody, které se volají podle datového typu proměnné, ze/do které se má zapisovat/číst. Odpovídají jim metody pro čtení `ReadInt`, `ReadDouble`, `ReadString` a `ReadBool`. Pro zápis jsou vytvořeny metody `WriteInt`, `WriteDouble`, `WriteString` a `WriteBool`.

3.3 MATEMATICKÉ KNIHOVNY

Watcom Fortran 77 využívá vnitřní funkce, které v C++ nenajdeme. Musel jsem proto vytvořit jejich ekvivalenty a to pro celočíselný a desetinný datový typ. Tyto funkce provádějí stejný algoritmický výpočet jako ve Fortranu.

LITERATURA

- [1] Schildt, H.: Nauč se sám C++, Praha 2001
- [2] Štěpánek, M.: Ročníkový projekt, Brno 2005
- [3] Watcom Fortran 77 – Language Reference. U.S.A., 2000