

A PARSING METHOD BASED ON A SET OF GRAMMARS

Josef MAJTÁN, Master Degree Programme (4)
Dept. of Information Systems, FIT, VUT
E-mail: xmajta00@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

This paper presents a parsing method based on a set of grammars and combines several parsing methods. Specifically, the present paper discusses a parsing method based on two grammars. The main method uses recursive-descent parsing. This method makes the parse of the whole body of the source code. The other method uses operator-precedence method to parse expressions. The resulting parsing method makes the parsing faster and more effective.

1 ÚVOD

Syntaktická analýza je součástí kompilačního procesu. Diskutovaná metoda zpracovává tokeny, které získává při volání lexikálního analyzátoru. Ten zpracovává zdrojový text po lexémech a vytváří jejich reprezentace, neboli tokeny. Syntaktická analýza prověřuje správnost zdrojového kódu z hlediska jeho skladby a musí se také umět zotavit z těchto chyb. Logické chyby však neodhalí, o ty se stará jiná část kompilátoru.

2 ROZBOR

Pro účely testování funkce diskutované metody jsem si vytvořil jednoduchý programovací jazyk. Jeho celá formální definice je uvedena v mém ročníkovém projektu [2]. Zde uvedu pouze některé podstatné znaky a vlastnosti, které tento programovací jazyk charakterizují.

2.1 PROGRAMOVACÍ JAZYK

- umožňuje deklaraci funkcí a procedur
- obsahuje podmíněný příkaz typu „if - then” nebo „if - then - else“
- také obsahuje rozšířený podmíněný příkaz typu „switch - case“
- dále obsahuje cykly typu „while - do“, „repeat - until“ nebo „for“
- umožňuje deklaraci jak lokálních tak globálních proměnných
- každý program musí obsahovat hlavní proceduru „main()“

Jak již bylo výše zmíněno, využívám v syntaktické analýze dvou metod, které se doplňují.

Hlavní metoda představuje rekurzivní sestup, což je metoda překladu shora dolů. Potřebuje pro svou činnost gramatiku typu LL1. Funguje tak, že ze startovacího nonterminálu začne pomocí prepisovacích pravidel generovat příslušnou větu programovacího jazyka. Které prepisovací pravidlo se má při derivování použít určuje token přicházející od lexikálního analyzátoru. Tato metoda řídí překlad celého zdrojového textu a ve vhodný okamžik se zastaví a zavolá pro překlad výrazů druhou metodu. Po jejím skončení pokračuje v překladu hlavní metoda.

Druhá metoda představuje precedenční syntaktickou analýzu, což je metoda překladu zdola nahoru. Pro svou činnost jí stačí bezkontextová gramatika. Levá rekurze zde není na překážku, ale právě naopak. Díky ní je gramatika ucelená a přehledná. Tato metoda postupuje tak, že postupně načítá tokeny z výrazu a pomocí opačně aplikovaných prepisovacích pravidel se pokouší celý výraz zredukovat na příslušný nonterminál.

Přepínání těchto dvou metod je určeno množinou dvojic stopových prvků. Princip je následující. Pokud hlavní metoda při překladu narazí na některý první stopový prvek z dvojice patřící do této množiny, pozná, že bude následovat výraz. Zastaví se a na zpracování výrazu zavolá druhou metodu. Tato metoda pokračuje v načítání tokenů a zpracování výrazu až do doby, než načte druhý prvek, patřící do dvojice s počátečním prvkem. Nyní metoda ví, že výraz je již celý načten. Musí dokončit redukci a následně předat řízení hlavní metodě.

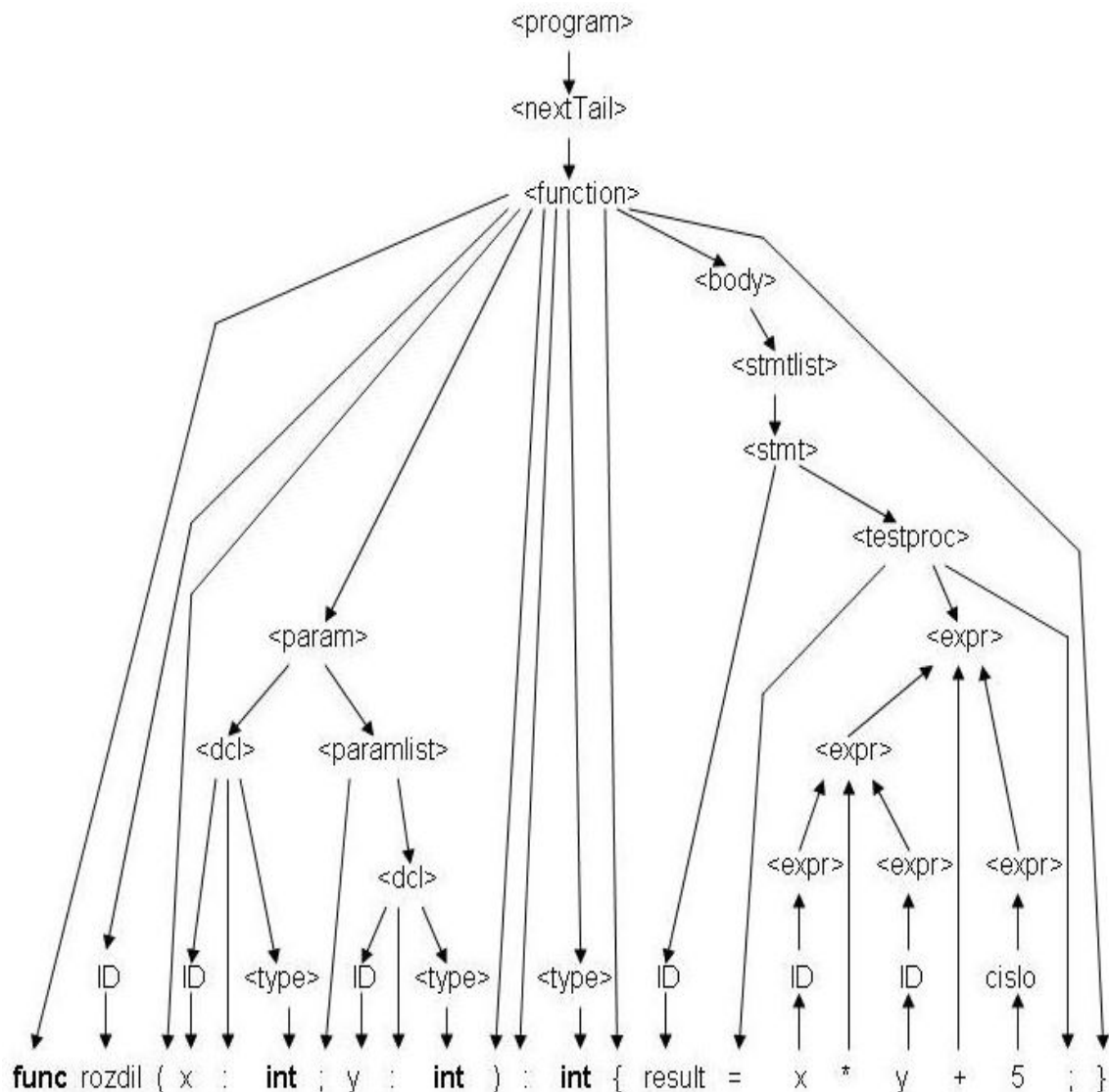
2.2 DVOJICE STOPOVÝCH PRVKŮ

- „if“ - „then“
- „case“ - „:“
- „while“ - „do“
- „(“ - „;“
- „;“ - „;“
- „;“ - „)“
- „=“ - „;“

Zotavení provádím na bázi metody pevných klíčů, kdy po zjištění syntaktické chyby dochází k přeskočení části zdrojového kódu až po prvek, který je stanoven jakožto klíčový a tedy zaručeně správný. Celý formální popis obou uvedených gramatik se nachází v mém ročníkovém projektu [2].

Výsledkem celého snažení je buď informační výpis, že zdrojový kód je syntakticky správný nebo výpis všech nalezených chyb s čísly řádků, na kterých se chyby vyskytovaly. Výsledným produktem tohoto projektu bude syntaktický analyzátor, který používá výše popsané metody. Jednou z oblastí jeho použití mohou být například kompilátory.

2.3 PŘÍKLAD



Obr. 1: *Derivace a redukce tvořící větu jazyka*

Obrázek 1 znázorňuje princip prezentované metody. Šipky, které směřují dolů, naznačují provádění postupné derivace a naopak šipky směřující nahoru znamenají redukce. Z obrázku je patrné, kdy předává hlavní metoda řízení překladače metodě druhé a naopak. Uzly stromu z obrázku jsou nonterminály, které reprezentují v implementaci použité rekurzivní funkce. Listy tohoto stromu, které jsou zvýrazněné tučným písmem označují klíčová slova programovacího jazyka.

LITERATURA

- [1] Meduna, A.: Automata and Languages: Theory and Applications, Springer, London, 2000
- [2] Majtán, J.: Syntaktická analýza založená na množině gramatik, [ročníkový projekt], VUT v Brně, FIT, 2005 (předběžná verze)