



Dokumentace k projektu pro předměty IFJ a IAL

Implementace interpretu imperativního jazyka IFJ12

Tým XXX, varianta x/x/x

Vedoucí týmu:

29. listopadu 2014

Řešitelé: 1BIT Průžina Tomáš, xpruzi01@stud.fit.vutbr.cz

Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1 Úvod

Implementácia prekladace imperatívneho jazyka je netriviálna úloha, preto je vhodné ju rozdeliť na podúlohy (kapitola 2), ktoré sú vhodné rozdelené medzi členy tímu, vedeného a kontrolovaného tímovým vedúcim.

Tento dokument sa skladá z N častí, ktoré popisujú jednotlivé moduly interpretu imperatívneho jazyka IFJ14, akožto lexikálny analyzátor (ref), syntaktický analyzátor (ref) využívajúceho rekurzívny sestup (ref), ktorý je pre potreby syntaktickej analýzy vyrazov rozšírený o Shunting Yard (ref) algoritmus. Yadi yadi yada...

2 Moduly interpretu jazyka

2.1 Lexikální analyzátor

Syntaktický analyzátor potrebuje ke svojej činnosti lexikálny analyzátor. Ten predáva na žiadosť SA tzv. tokeny, ktoré získava postupným čítaním vstupného suboru, ktorý obsahuje zdrojový kód napísaný v jazyku IFJ14. Samotný výsledný token je reprezentáciou lexemu (identifikátor, kľúčové slovo, príkaz priradenia atp.).

Pre úspešné rozlíšenie typu lexemu sa v našej implementácii používa konečný automat, ktorého štruktúra je znázornená obrázkom (ref).

Ak sa konečný automat počas spracovávania reťazca zdrojového suboru dostane do chybového stavu `lexerref`, reťazec je neprijateľný a jedná sa o lexikálnu chybu. Samotná implementácia lexikálneho analyzátoru patrí k jednoduchším častiam projektu, avšak jeho návrh a testovanie zabral nemalo času (liez).

insert KA diagram here

2.2 Syntaktický analyzátor

2.2.1 Precedenční syntaktický analyzátor

2.3 Interpret

Interpret je záverečnou časťou projektu. Rekurzívne spracováva abstraktný syntaktický strom reprezentovaný binárnym stromom a postupne vykonáva operácie na ňom definované.

Koreňom tohto AST je vždy typ prevádzanej operácie, pričom obsahuje predom definované nepovinné parametre (príkladom je podstrom volania funkcie, ktorý obsahuje predávané premenné). Teda každý uzel AST predstavuje jednu atomickú operáciu, ktorou je riadený samotný beh programu.

Interpret samozrejme nepracuje len so samotným syntaktickým stromom, ale taktiež s tabuľkou symbolov, ktorá obsahuje dáta potrebné na to, aby interpret mohol vykonávať nejakú užitočnú prácu definovanú v tele programu.

Interpret teda spravuje tabuľky symbolov, dynamicky ich za behu vykonávaného programu vytvára a (podľa potreby) ruší.

V neposlednej rade interpret vykonáva semantickú kontrolu pri operáciách, ktoré nie je syntaktický analyzátor v dobe spracovávaného vstupného programu a generácie abstraktného syntaktického stromu riešiť.

Prikladom takejto kontroly je pouzivanie (citanie) premennej pred jej riadnou definicou (priradenim hodnoty) alebo delenie nulou v aritmetickom vyraze.

Taketo behove chyby su v interprete riesene ukoncenim interpretacia a navratom chyboveho kodu podla zadania imperativneho jazyka IFJ14.

2.3.1 Tabulka symbolů

Tabulka symbolov je implementovana pomocou binarneho vyhľadavacieho stromu, pricom sa rozlisuje niekoľko urovni tabulky symbolov.

Kazdy program obsahuje globalnu tabulku symbolov (v ktorej su ulozene ukazatele na funkcie) a lokalnu tabulku symbolov, ktora obsahuje behove premenne vykonavaneho programu.

Interpret spravuje tabulk

Z implentacneho hladiska su lokalne tabulky symbolov (stromy) ukladane na zasobnik, pricom plati, ze pri volani funkcie sa vytvori nova (lokalna) tabulka symbolov, do ktorej sa skopiruju prislusne parametre volanej funkcie z nizsej vrstvy lokalnej tabulky (popr. z globalnej tabulky symbolov).

Takato lokalna tabulky taktiez obsahuje navratovu hodnotu vykonavanej funkcie, co umoznuje rekurzivne volanie funkcii a samozrejme umoznuje vykonavat telo funkcie bez nutnosti akokolvek zasahovat do abstraktného syntaktického stromu.

3 Postup při implementaci řešení

4 Závěr

A Metriky kódu

Reference

- [1] HONZÍK J. M.: *Studijní opora pro předmět Algoritmy*. Elektronický text. FIT VUT v Brně
- [2] MEDUNA A., LUKÁŠ R., *Podklady k přednáškám*. Elektronický text. FIT VUT v Brně