

Implementační dokumentace k 1. úloze do IPP 2023/2024

Jméno a příjmení: Michal Blažek

Login: xblaze38

Popis programu

Tento program slouží jako překladač jazyka IPPcode24 do výsledného XML formátu. Program je napsán v jazyce Python a zpracovává vstupní soubor ze standardního vstupu, provádí lexikální a syntaktickou analýzu a zapisuje výsledný XML soubor na standardní výstup.

Dekompozice projektu

Projekt se dá rozdělit na 4 hlavní části (zpracovávání argumentů, lexikální analýza, syntaktická analýza a generování kódu).

Zpracování vstupních argumentů programu

Program zpracovává data na standardním vstupu a vypisuje na standardní výstup, tudíž nepotřebuje žádné další argumenty pro specifikování například cesty ke zdrojovému souboru. Jediný argument, který program zpracovává, je `-h`, nebo jeho delší verzi `--help`. Tento argument vypíše nápovědu k programu.

Lexikální analýza

Další částí je lexikální analýza. Lexikální analýza nejprve najde v kódu znaky `#` a oddělí je od ostatních znaků pomocí mezery před i za znakem `#`. Tím se můžou v kódu vyhledat jednořádkové komentáře a odstranit. Dalším úkolem lexikální analýzy je rozdělení vstupního souboru na tokeny, reprezentující jednotlivé instrukce a jejich argumenty. Těmto tokenům pak přiřadí typ podle jejich hodnoty.

Syntaktická analýza

Na začátku syntaktické analýzy je zkontrolováno, zda první instrukce v programu je hlavička jazyka IPPcode24. Zbytek syntaktické analýzy je implementován pomocí rekurzivního sestupu založeného na LL gramatice, která je přiložena na konci tohoto souboru.

Funkce `prog()` rozděluje vstupní soubor na řádky ukončené znaky `\n` nebo koncem souboru. Pro zpracovávání prázdných řádků jsou v LL gramatice využity neterminály `enter` a `another_enter`.

Funkce `command()`, která reprezentuje stejnojmenný neterminál, kontroluje jednotlivé řádky zdrojového kódu a následně podle názvu instrukce řeší její argumenty.

Zbylé funkce slouží pro zpracování argumentu instrukce podle jejího typu.

Generování výsledného XML kódu

Během generování kódu je zapotřebí vytvořit hlavičku XML souboru. Poté se vypíše štítek `<program>` s předepsanými argumenty. Pro každou instrukci v IPPcode24 vytvoříme štítek `<instruction>` a do něj jsou vnořeny `<arg1|2|3>` štítky. Všechny štítky mají nějaké argumenty, ve kterých jsou uložena data ze vstupu, která se úspěšně zpracovala a prošla lexikální i syntaktickou analýzou.

Pro generování XML kódu existují funkce `headerGen()`, `programGen()`, `instructionGen()` a `instrArgGen()`. Tyto funkce se volají během syntaktické analýzy pokaždé, když je daná část zpracována a označena za korektní.

Při generování XML hlavičky jsem narazil na problém s přidáním atributu `encoding="UTF-8"`. Tento problém je řešen umělým přidáním řádku se správnou hlavičkou a následným odstraněním nevhodného řádku z výsledné XML reprezentace kódu.

LL gramatika

1. <prog> -> <command> <enter> <prog>
2. <prog> -> \0
3. <enter> -> \n <another_enter>
4. <another_enter> -> <enter>
5. <another_enter> -> epsilon
6. <command> -> COMMAND
7. <command> -> epsilon
8. <var> -> GF|LF|TF
9. <symb> -> GF|LF|TF|BOOL|STRING|INT|NIL
10. <label> -> LABEL|DATA_TYPE|COMMAN
11. <data_type> -> DATA_TYPE

Tučně zvýrazněné jsou typy tokenů.