

Implementačná dokumentácia k 2. úlohe do IPP 2019/2020

Meno a priezvisko: Martin Fekete

Login: xfeket00

interpret.py

Spracovanie argumentov a parsing XML

Na začiatku programu sú skontrolované argumenty programu a zároveň načítaný zdrojový súbor či pomocou štandardného vstupu, alebo zadaného súboru, pričom celý jeho obsah je uložený do globálnej premennej `input_file` (obdobne je spracovávaný súbor so vstupom pre inštrukciu *READ*, ktorý je ale spracovávaný až po zavolaní tejto inštrukcie).

Po uložení obsahu zdrojového súboru je jeho obsah sparovaný pomocou funkcie `fromstring` z knižnice `ElementTree`. Ak pri parsingu nastane chyba, znamená to, že zdrojový kód nie je *well-formed* a program teda končí s chybou 31. Inak je skontrolovaná syntaktická správnosť koreňových elementov zdrojového súboru a obsah premennej obsahujúcej sparovaný XML súbor je zoradený najprv podľa poradia *order*, po čom sú zoradené aj *argumenty* inštrukcií do správneho poradia. Počas zoradovania je takisto kontrolovaná syntaktická správnosť inštrukcií.

Návestia

Keďže v `ippcode20` je možné prevádzať skoky aj na zatiaľ nedefinované návestia, XML je prechádzané v 2 priechodoch. V prvom priechode sú uložené mená všetkých návestí spolu s indexom, na ktorom sa nachádzajú a takisto sú uložené všetky inštrukcie do poľa inštrukcií `instructions`. V druhom priechode je toto pole prechádzané v “nekonečnom” cykle, pričom je vždy volaná funkcia `interpret` s argumentom, ktorý obsahuje index inštrukcie v poli `instructions`. Počítadlo indexu do tohto poľa je každým novým cyklom inkrementované, prípadne je zmenené v inštrukciách skoku.

Pamäťový model

Globálny a lokálny pamäťový rámec sú implementované ako slovníky s názvami `GF`, `LF` alebo `TF`, do ktorých sú premenné ukladané vo formáte *‘názov’ : ‘hodnota’*. Lokálny rámec je implementovaný ako pole slovníkov, do ktorých sú premenné ukladané obdobne. Dátové typy premenných sú ukladané v slovníkoch s názvami `GF_type`, `LF_type` alebo `TF_type` v podobnom formáte. Na kontrolu, či je dostupný dočasný rámec je použitá premenná typu `bool` `TF_access`, dostupnosť lokálneho rámca je kontrolovaná zásobníkom `pushed`, ktorý obsahuje hodnoty `true` alebo `false`.

Funkcia interpret

Funkcia `interpret` je v podstate jeden *if-else* blok, ktorý na základe operačného kódu aktuálnej inštrukcie zavolá funkciu s rovnomeným názvom, ktoré už obsluhujú syntaktickú a sémantickú kontrolu a samozrejme samotnú interpretáciu kódu.

Za zmienku stojí funkcia `iterate_var_symb`, ktorá je využívaná pri inštrukciách tvaru `OPCODE <var> <symb1> [<symb2>]` a jej úlohou je zistiť, do akej premennej bude výsledok inštrukcie ukladaný (spolu so sémantickými kontrolami) a ďalej vráti hodnotu a typ uložený v `symb1` a `symb2`. Toto výrazne zjednodušuje a unifikuje prácu s premennými a konštantami.

Ukončenie interpretácie

Interpretácia je ukončená buď v prípade chyby alebo v prípade, že hodnota počítadla indexu `curr_instr` do poľa `instructions` je väčšia ako dĺžka tohto poľa.

test.php

Spracovanie argumentov

Keďže skript môže dostať pomocou argumentov až 3 rôzne súbory, sú pre tento účel vytvorené globálne premenné, do ktorých je na začiatku uvedená implicitná cesta k súboru podľa zadania. V prípade, že používateľ nejaký z týchto súborov zadá pomocou argumentu, jednoducho sa prepíše implicitná cesta tou, ktorú zadal používateľ v argumente. Argumenty sú potom jednoducho spracované pomocou `foreach` cyklu.

Prepínač `--recursive`

Prepínač `recursive` je riešený tak, že na začiatku programu je do globálnej poľa `dir` uložená cesta k aktuálnemu adresáru, v prípade použitia prepínaču `--directory` je cesta prepísaná cestou zadanou používateľom. Ak bol použitý prepínač `--recursive`, sú vo funkcii `recursiveDirectories` rekurzívne nájdené podadresáre adresáru uloženého v poli `dir[0]` a do toho istého poľa sú aj uložené. Toto zjednocuje prácu s funkciami `parseOnly`, `intOnly` a `both` tak, že vždy je možné prechádzať pole `dir` položku po položke a tú “posielať” do týchto funkcií.

Režim `--parse-only`

Pri použití tohto prepínaču je zavolaná funkcia `parseOnly`, ktorá najprv vyhľadá všetky súbory končiace príponou `.src` a uloží ich do poľa `srcFiles`. Pole je potom prechádzané po jednej položke a to tak, že vždy je pomocou funkcie `exec` spustený príkaz `php7.4 $parseScript < $srcFilePath > xfeket00-test`, kde:

- `$parseScript` je cesta k súboru `parse.php`
- `$srcFilePath` je cesta k aktuálnemu testovanému `.src` súboru
- `xfeket00-test` je dočasný súbor, do ktorého je ukladáný výstup parseru

Ďalej je vyhľadaný súbor s príponou `.rc` a rovnakým názvom ako aktuálny `.src`. Ak tento súbor nie je nájdený, je vytvorený a je doňho zapísaná hodnota 0, s ktorou sa ďalej pracuje. Ak nájdený je, je jeho obsah uložený do premennej `rcContents`. Ak je hodnota premennej `retVal` (v ktorej je uložený návratový kód z parseru) iný ako 0, je táto hodnota porovnávaná s hodnotou uloženou v `rcContents`. Inak je porovnávaný ako aj návratový kód, tak aj obsah súboru `xfeket00-test` a `.out` súboru s rovnakým menom ako `.src` pomocou nástroja `JExamXML` (ktorého cesta je v premennej `jexamxml`). Ak prebehnú všetky kontroly správne, je zavolaná funkcia `generatePass` na vygenerovanie výpisu prejdeného testu do HTML. Inak je zavolaná funkcia `generateFail`. Súbor s príponou `.in` sa pri `--parse-only` ignoruje.

Režim `--int-only`

V prípade prepínača `--int-only` je zavolaná funkcia `intOnly`, ktorá je takmer identická s funkciou `parseOnly`. Jediné rozdiely sú, že funkcia pomocou `exec` spúšťa príkaz `python3.8 $intScript --source=$srcFilePath --input=$inFilePath > xfeket00-test`, kde `$inFilePath` je súbor s obsahom `.in` súboru (prípadne prázdny `.in` súborom, ak predtým neexistoval). Výstup uložený v `xfeket00-test` je potom porovnávaný s `.out` súborom pomocou nástroja `diff`.

Režim `both`

V prípade režimu `both` je zavolaná funkcia `both`, ktorá je v podstate iba kombináciou funkcií `intOnly` a `parseOnly`. Najprv je súbor s príponou `.src` “poslaný” do parseru, ktorého výstup je posunutý interpretu.

Výpis do HTML

V prípade všetkých 3 režimov sú dočasne vytvorené súbory zmazané na konci jednotlivých funkcií. Zápis na štandardný výstup vo formáte HTML je riešený pomocou funkcií začínajúcich na `generate` pomocou príkazu `echo`.