Implementačná dokumentácia k 2. úlohe do IPP 2019/2020

Meno a priezvisko: Dávid Oravec
Login: *xorave05*

1 interpret.py

Interpret najprv spracuje argumenty pomocou funkcie `argHandler()`, ktorá využíva vstavanú knižnicu Pythonu `argparse` a vráti vstupný súbor s XML reprezentáciou kódu, súbor so vstupmi pre interpretáciu a prípadne súbor pre zápis štatistík. Pokiaľ nie je zadaný súbor s XML reprezentáciou kódu alebo súbor so vstupmi pre interpretáciu, tak sa dáta čítajú zo `sys.stdin`. Následne je volaná funkcia `readSource()`, ktorá opäť pomocou vstavanej knižnice `xml.etree.ElementTree` rozoberie XML reprezentáciu kódu a skontroluje jeho správnosť. Na záver sa zoradia príslušné inštrukcie podľa poradia v XML kóde (order) a funkcia vráti `dictOfInstructions` vo formáte `{'order': ['opcode', [arg1 , ...]], ...}`, kde arg je dvojica `typ, hodnota`. Ešte pred samotnou interpretáciou kódu sa vytvoria všetky náveštia pomocou funkcie `createLabel`, ktoré sa nachádzajú v slovníku inštrukcii.
Na začiatku interpretácie je vytvorená `hashTable["GF"]` a `hashTable["label"]` čo sú v podstate opäť slovníky, jeden vytvára Globálny rámec a druhý vytvára miesto na uchovávanie indexu a mena náveštia. Inštrukcie sa načítavajú po jednom vo while cykle v ktorom je využitá funkcia `mySwitch()`, ktorá imituje funkcionalitu bežného switch ako napríklad v jazyku C. Na základe spracovania inštrukcie v tomto switchi je volaná odpovedajúca funkcia. Každá jednotlivá inštrukcia ma odpovedajúcu funkciu v ktorej je vykonávaná.

1.1 Rozšírenia

Skript podporuje rozšírenia STACK, FLOAT a STATI.

1.1.1 STACK

Rozšírenie STACK je implementované veľmi podobne ako implementácia nezásobníkových inštrukcii. Jednotlivé symboly su vyťahované zo zásobníka pomocou `stackOfVars.pop()` a vkladané na zásobník pomocou `stackOfVars.append()`, kde `stackOfVars` je typu list().

1.1.2 FLOAT

Rozšírenie FLOAT je implementované pridaním nového podporovaného datového typu, ktorý musel byť do jednotlivých funkcii pridaný. Pri prevádzaní sú používané funkcie `float.fromhex()` a `float.hex()`.

1.1.3 STATI

Rozšírenie STATI spočíva v zbieraní štatistík, ktoré sú spočítavané v hlavnom tele programu. Pri počítaní inštrukcii robili najväčší problém skokové inštrukcie. Počítanie premenných prebieha jednoduchým cyklom, ktorý prebehne cez všetky rámce, napočíta aktuálny počet definovaných premenných a následne pomocou funkcie `max()` zistí či je aktuálny počet premenných vačší ako pôvodný a uloží nový maximálny počet do globálnej premennej.

2 test.php

Skript začne s kontrolou argumentov vo funkcii `argHandler()`. Následne je vykonávaná funkcia `directoryCheck()`, ktorá skontroluje jednotlivé zložky, či existujú a či sú to validné zložky. Funkcia vracia pole s cestami súborov. Potom je toto pole poslané do funkcie `sortTestsToArray()`, kde sa k príslušným src súborom vytvorí asociatívne pole s indexami `src, out, in, rc`. Takto zoradené jednotlivé polia sú potom pridané do finálneho pola, ktoré je z funkcie vrátené. Po vytvorení pola s testami sa jednotlivé testy začnú vykonávať a podľa zadaných prepínačov porovnávať výsledky testov buď pomocou nástroja JExamXML alebo pomocou diff. Na záver je na štandradný výstup vypísaný HTML dokument so zoznamom jednotlivých testov a ich výsledkov a prehľad o počte úspešných a neúspešných testov z celkového počtu.