Implementační dokumentace k 1. úloze do IPP 2023/2024

Jméno a příjmení: Martin Vrablec

Login: xvrabl06

Stručná dokumentácia o skripte `parse.py` a jeho funkcionalite

`main()` – Hlavná funkcia v ktorej, sa načíta z `stdin` daný súbor zbavý sa komentárov a zbývajúce časti ktoré su odelené medzerníkom sa uložia do listu následne sa list prechádza a elementom listu sa určuje typ podľa lexikálnej analýzy pomocou funkcie `set_type()` do ktorej sa posiela premmena typu `Token` potom sa list s určenými typmi prechádza funkciou `parser()` z ktorej si ukladáme návratový typ pre prípadnú chybu v syntaktickej analýze.

Lexikálna analýza

`set_type(token)` – V tejto funkcii sa nastavuj typ prichádzajúceho tokenu podľa regex výrazov alebo podľa presne určených vyhradených slov. Na opcodes je využitá metóda `.upper()` nakoľko su case insensitive. Funkcia nič nevracia len modifikuje premenné typu `Token`.

Syntaktická analýza

`parser(token_array)` – Funkcia . Najskôr funkcia zisťuje správnosť poprípade absenciu hlavičky, následne sa vytvorý koreňový element xml štruktúry. Ako ďalší krok sa zoberie prvá token listu ktorý by mal byť opcode alebo `EOL` tie sú potom nájdené slovníku `instruction_dict` (ak nie vraciame príslušný chybový kód) do premennej `expected_tokens` sa uložia typy tokenov ktoré, by daný opcode mali následovať. Vytvorý sa xml štruktúra pre danú inštrukciu pomocou funkcie `instruction_to_xml()` a začne sa prechádzať list `expected_tokens,` typ tokenov z pôvodného listu (ten je určený pomocou funkcie `return_type()`)je porovnávaný typom v expected tokens na danej pozícií taktiež sa kontroluje aj či má danný typ podtriedu, ktorá je tiež validným typ. Ak tieto kontroly prejdú vytvára sa xml element pre argumenty inštrukcie `argument_to_xml()` ak sa prejde celý list `expected_tokens` prechádza sa na daľšiu inštrukciu. Po prejdení celého poľa zo vstupu sa vytvorí finálna xml štruktúra pomocou funkcie `format_xml()` a vypíše sa na `stdout` . Funkcia vracia 0 ak chyba nenastala inak vracia príslušne chybové návratové čísla.

`return_type(type)` – Funkcia načíta reťazec `type` podľa, ktorého určí daný typ `Var, Symb, Type, Label, EOL, Unk.` Funkcia vracia typ.

Generovanie XML

`instruction_to_xml(root,order,type)` – Funkcia pridáva danému `root` elemntu podelement `instruction` a nastavuje atribúty pre order na `order` a opcode na `type,` pre `EOL` necheme aby bol daný element vytvorený. Funkcia vracia `order` a `instruction` pre generovanie daľších inštrukcií, so správnym poradám a pre správne priradenie argumentov vo funkcií `argument_to_xml()`.

`argument_to_xml(instruction,type,data,position)` – Funkcia pridáva danému `instruction` elementu podelemnt arg nastavuje atribúty typ na `type` a text na `data,` určuje poradie argumentu pomocou `position,`pre `EOL` necheme aby bol argument generovaný. Funkcia nič nevracia.

`format_xml(root)` – Funkcia prevedie daný xml strom `root` pomocou `ET.tostring()` (z modulu ElementTree) do reťazca, ktorý postupne formátuje pomocou `toprettyxml()` (z modulu minidom). Funkcia vracia formátovanú štruktúru xml.

Classy - `Var, Symb, Type, Label, EOL, Unk` sú vytvorené len na určenie typu daného token a na vytvorenie slovníku `instruction_dict` pre syntaktickú analýzu. `Token` je vytvorený na lepšiu prácu s tokenmi, v určitých prípadoch ja za potrebie typ daneého tokenu a zároveň jeho hodnota. `Types` je vytvorený pre získanie typov v slovníku `instruction_dict.`