Implementační dokumentace k 1. úloze do IPP 2018/2019
Jméno a příjmení: Ondřej Studnička
Login: xstudn00

Skript parse.php:

Úkolem bylo vytvořit skript parse.php, který provede lexikální a syntaktickou analýzu jazyka IPPcode19 a následně jej převede do XML reprezentace v udané formě. Skript je vypracován v jazyce PHP, konkrétně ve verzi 7.3.

Implementace:

V první řadě bylo třeba zajistit správné spuštění skriptu uživatelem. Skript lze spustit následujícími způsoby: „`php7.3 parse.php < vstupni_soubor`“ / „`php7.3 parse.php --help`“. Pokud je skript spuštěn, tak jako je ukázáno v první variantě, tak může začít vykonávat svůj hlavní úkol. Druhý z uvedených způsobů vypíše uživateli nápovědu k používání. Jiné spuštění skript ukončí s chybou 10.

Ze vstupního souboru se ve smyčce načítají řádky. Pokud načtený řádek obsahuje sekvenci bílých znaků, pak se tyto znaky sloučí do jedné mezery. Pokud načtený řádek obsahuje komentář (sekvence znaků za #), pak se tyto znaky (včetně #) nahradí prázdným řetězcem. Takto upravené řádky skript ukládá do pole `$array_of_lines`.

V dalším kroku skript kontroluje, zda se na prvním načteném řádku nachází hlavička ve tvaru `.IPPcode19`. Když tato podmínka není splněna, pak parse.php končí s chybou 21. Pokud je vše v pořádku, pak se z pole načtených řádků hlavička oddělá. Hlavička může být zadána, jak malými, tak velkými písmeny.

Dále následuje vytvoření osmi polí a proměnných obsahující regulární výrazy. Pole obsahují vzorové instrukce a jsou logicky členěna podle toho, jestli jsou v nich uloženy bezoperandové instrukce, jednooperandové instrukce, po kterých následuje proměnná/návěští/symbol atd. Do regulárních výrazů se ukládá vzor, který bude v následující části sloužit ke kontrole neterminálů .

Pro správné pokračování skriptu je potřeba zajistit rozdělení každého řádku na jednotlivá slova. Za tímto účelem využívá parse.php funkci `explode`, která každý řádek v místě mezery rozseká do slov a ty následně uloží do pole `$word`. Ještě před začátkem hlavní kontroly instrukcí a operandů se parse.php dívá na každé vzniklé slovo a zjišťuje, jestli se v něm nachází zpětné lomítko. Pokud ano, tak za lomítkem musí následovat sekvence tří čísel, jinak se jedná o špatný zápis a proces končí s chybou 23. Pokud je vše v pořádku, pak se pokračuje k hlavní kontrole instrukcí a operandů. Hlavní kontrola probíhá pro každé pole `$word` následovně. Z pole si skript vezme nultý prvek (instrukce) a zkontroluje, zda se nachází v jednom z osmi výše popsaných polí. Pokud se prvek v polích nenachází, tak parse.php končí s chybou 23. V opačném případě se přistoupí ke kontrole operandů (pokud je zkontrolovaná instrukce bezoperandová, tak je rovnou uložena do pole `$array_of_checked_instructions`). Skript již ví, v jaké kategorii se generovaná instrukce nachází, tím pádem ví, kolik očekávat operandů. Parse.php vypočte velikost pole `$word` a porovná ji s očekávanou velikostí. Pokud se čísla rozcházejí, pak je skript ukončen s chybou 23. V opačném případě přistoupí ke kontrole samotných operandů za pomoci regulárních výrazů. Zadané operandy jsou porovnávány s regulárními výrazy pomocí funkce `preg_match`. Pokud je mezi operandem a regulárním výrazem nalezena shoda, pak je vše v pořádku, řádek je tedy dostatečně otestován a skript jej uloží do pole `$array_of_checked_instructions`. Pokud shoda nalezena není, pak operand nemá požadovaný tvar. Chyba 23. Po kontrole všech řádků pokračuje skript ke generováním XML reprezentace.

Generování je zajištěno pomocí XMLWriter. V první řadě je vygenerována hlavička obsahující informace o verzi XML a kódování. Za hlavičkou se vyskytuje tag `<program>` nesoucí informaci o jazyce. Následuje generování jednotlivých instrukcí a operandů. Z pole otestovaných instrukcí bere parse.php řádek po řádku a na základě testování o jaký typ instrukce se jedná (bezoperandová, …) generuje patřičný XML kód pro instrukce a jejich operandy. V posledním kroku je XML kód vypsán na standardní výstup.