

# Implementačná dokumentácia k 1. úlohe do IPP 2020/2021

Meno a priezvisko: Jakub Bartko

Login: xbartk07

## 1 Štruktúra kódu

Činnosť skriptu `parser.php` je možné rozdeliť na spracovanie argumentov príkazového riadka, načítavanie zdrojového kódu, lexikálnu a syntaktickú analýzu inštrukcií, generovanie výstupu vo formáte XML a zbieranie štatistík o zdrojovom kóde. Hlavné súčasti týchto činností sú rozdelené na niekoľko hlavných funkcií, ktoré ďalej využívajú pomocné obslužné funkcie a štruktúry.

Medzi tieto štruktúry patria:

- trieda **State** — jednoduchý zoznam stavov konečného automatu, ktorým je parser riadený
- trieda **Counters** — počítadlá riadkov kódu, komentárov, ... a pomocné zoznamy návěstí; slúžiace na zber štatistík pre rozšírenie STATP
- asociatívne pole **opcodes** — zoznam platných inštrukcií zdrojového kódu a typov ich argumentov (**variable**, **symbol** alebo **label**)
- asociatívne pole **opts** — zoznam zadaných súborov štatistík a príslušných cieľových súborov

Okrem toho je výstup vo formáte XML zbieraný v premennej `[string] outputs` v rôznych bodoch parsovania.

## 2 Činnosť parsera

**Argumenty príkazového riadka** sú spracované iterovaním, kontrolou správnosti s využitím regulárnych výrazov a generovaním asociatívneho poľa **opts** pre príslušné súbory štatistík.

**Parsovanie zdrojového kódu** je v základe štruktúrované ako primitívny konečný automat. V každom svojom cykle načíta jeden riadok zo štandardného vstupu, odstráni z neho komentáre a riadok preskočí, ak obsahuje výhradne biele znaky. Ako prvý neprázdny riadok musí spracovať hlavičku zdrojového kódu; ďalej spracováva inštrukcie a ich argumenty.

**Spracovanie inštrukcie** pozostáva z kontroly jej operačného kódu a kontroly počtu a typu jej argumentov podľa referenčného asociatívneho poľa **opcodes**, a generovania XML elementu **instruction**. Obidve z uvedených kontrol sú vykonané s využitím regulárnych výrazov na základe definovaného typu: **variable**, **symbol** alebo **label**. Pri type **variable** sa skontroluje platnosť označenia rámca a formát identifikátora; pri type **symbol** sa skontroluje možnosť uvedenia premennej (kontrola identická s typom **variable**) a uvedenia konštanty, pri ktorej sa skontroluje jej špecifikovaný typ a platnosť príslušnej hodnoty. Okrem toho sa v reťazci, ktorý predstavuje argument inštrukcie, nahradia problémové znaky (pre formát XML) za zodpovedajúce XML entity.

## 3 Rozšírenie STATP

Štatistiky o zdrojovom kóde sú zbierané na príslušných miestach v skripte — napr. počet riadkov kódu pri prečítaní nového neprázdneho riadku zdrojového kódu zo štandardného vstupu, alebo počet definovaných návěstí pri parsovaní inštrukcie s operačným kódom **LABEL**. Dodatočné spracovanie si však vyžaduje vyhodnotenie počtu rôznych typov skokov (skokov dopredu, dozadu, alebo neplatných skokov). Toto vyhodnotenie je zabezpečené udržiavaním *zoznamu definovaných návěstí* a *zoznamu destinácií skokov*. Pri inštrukcií s operačným kódom **LABEL** sa definované návestie uloží do prvého z týchto zoznamov a v prípade, že sa vyskytuje v zozname druhom, pripočíta sa ako *skok dopredu*. Analogicky sa destinácia skoku pripočíta ako *skok dozadu*, ak pri inštrukcií podmieneného alebo nepodmieneného skoku špecifikované návestie už nachádza v zozname definovaných návěstí. Nájdene cieľové návestia sa z príslušného zoznamu odstránia a tie, ktoré v ňom zostanú po prejdení celým zdrojovým programom sú považované za **bad jumps**. Na záver parsovania je zoznam **opts** iterovaný a špecifikované súbory štatistík sú zapísané do príslušných súborov.