Implementačná dokumentácia k 1. úlohe do IPP 2024/2025
Meno a priezvisko: Patrik Procházka
Login: xprochp00

1 Úvod

Program `parse.py` typu filter, načíta vstupný program v jazyku SOL25 zo štandardného vstupu a na štandardný výstup vypíše XML reprezentáciu abstraktného syntaktického stromu. V prípade použitia argumentu `-h/--help` program vypíše krátky popis programu.

2 Analýza vstupného programu

2.1 Lexikálna a syntaktická analýza

V rámci lexikálnej a syntaktickej analýzy vstupného programu bola použitá knihovna `Lark`. Po zadefinovaní gramatiky podľa zadania, vo forme ENBF a následnom volaní funkcie `parse` prebehne kontrola, ktorej výsledkom je strom použitých *gramatických pravidiel* a *terminálov*. V prípade lexikálnej alebo syntaktickej chyby je odchytená a spracovaná výnimka vyvolaná touto funkciou. Terminály sú definované pomocou **regulárnych výrazov** a rekurzívne gramatické pravidlá sú zapísané vo forme **opakovacích pravidiel**, ktoré predchádzajú nežiadúcemu zanorovaniu rekurzívnych pravidel v syntaktickom strome.

2.2 Abstraktný syntaktický strom

Syntaktický strom vytvorený funkciou `parse` je následne transformovaný na *abstraktný syntaktický strom* štruktúrou podobný formátu JSON a to využitím triedy `Transformer` podľa transformačných funkcií definovaných pre gramatické pravidlá.

2.3 Sémantická analýza

Sémantické kontroly sú vykonávané v rámci prechodu cez AST, pred generovaním jednotlivých elementov výslednej XML reprezentácie. Na kontrolu zasielania **triednych správ** a ich porozumeniu je definovaná funkcia `check_class_message`, ktorá overí, či daný selektor správy patrí medzi triedne metódy. V prípade zaslania triednej správy `read`, je volaním funkcie `is_subclass` overené, že daná trieda je (pod)triedou `String`, a teda rozumie tejto správe. Medzi ďalšie funkcie, ktoré využívajú slovník `user_classes`, do ktorej je pri prvom prechode AST uložená informácia o rodičovskej triede, patrí funkcia s názvom `check_cyclic_inheritance`, ktorá pri definícii triedy, skontroluje potenciálnu **cyklickú dedičnosť** s jej nadtriedou. Pri väčšine sémantických kontrolách sú využívané definované zoznamy reťazcov, ako napríklad zoznam *kľúčových slov*, *vstavaných tried*, *globálnych objektov* a *pseudopremenných*.

3 Generovanie XML

Poslednou fázou programu je generovanie XML elementov vužitím knižnice `xml.etree.ElementTree` Funkcia `generate_xml` vytvorí koreňový element `program` a predá tento rodičovský element funkcii `generate_class`, ktorá vytvorí element `class` a volaním ďalších funkcií sa generujú jednotlivé elementy triedy, týmto spôsobom z **vrchu nadol** sa vygeneruje celý XML výstup.

Pred vypísaním výstupu je nad *koreňovým elementom* zavolaná funkcia `format_xml`, ktorá nastaví odsadenie elementov, kódovanie a pridá hlavičku dokumentu.