Praktická aplikácia regulárnych výrazov. Problémy a ťažkosti pri ich písaní a používaní.*

Denis Danilov

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xdanilovd@stuba.sk

2. novembra 2023

Abstrakt

Regulárne výrazy, často známe ako regex, je formálny jazyk, ktorý sa používa na priraďovanie textových vzorov. Používanie regexov pomáha rýchlo a presne nájsť a extrahovať kľúčové informácie z veľkých objemov textu. Môžu Vám doslova pomôcť nájsť ihlu v kope sena, čo organizáciám umožňuje presmerovať svoje ľudské zdroje do iných oblastí. Regexy sú obzvlášť užitočné na definovanie filtrov, pretože obsahujú postupnosť znakov, ktoré definujú vzor textu, ktorý sa má zhodovať.

Cieľom tejto práce je preskúmať, ako sa regulárne výrazy často používajú v praxi, identifikovať problémy a ťažkosti, s ktorými sa programátori stretávajú pri písaní a používaní regulárnych výrazov.

Metódami na dosiahnutie cieľa bude analýza prieskumu medzi vývojármi a údajov o používaní regulárnych výrazov v otvorených projektoch na GitHub.

Tento článok jasne ukáže výhody používania regulárnych výrazov, ako aj osvedčené postupy pri ich písaní.

1 Úvod

Regulárne výrazy, tiež známe ako Regexy, sú nástrojom na spracovanie textu, ktorý je integrovaný do všetkých súčasných programovacích jazykov a bežne používaných utilít, ako sú textové editory. Regulárne výrazy sa používajú pri úlohách vyhľadávania a nahrádzania reťazcov, ako je vyhľadávanie slov, úprava textu a analýza súborov. [?] Odhady naznačujú, že viac ako tretina projektov JavaScript a Python obsahuje aspoň jeden regulárny výraz. [?] Pochopenie a písanie regulárnych výrazov si vyžaduje znalosti aj zručnosti. [?] [?] Jediná chyba v znaku môže spôsobiť drasticky odlišné správanie regulárneho výrazu pri porovnávaní. [?] However, over 80% of regular expressions written in GitHub projects are not tested [?], indicating that developers either do not test regular expressions or use external tools rather than test cases. [?] Tento článok je určený na to, aby vám pomohol pochopiť a efektívne používať regulárne výrazy.

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2023/24, vedenie: PaedDr. Pavol Baťalík

Začneme základnými pojmami a postupne prejdeme k pokročilejším príkladom. Regulárne výrazy sú široko používané v rôznych oblastiach výpočtovej techniky. Tu je niekoľko bežných prípadov použitia:

1. Analýza protokolov: Regulárne výrazy možno použiť na extrahovanie špecifických častí informácií zo systémových alebo aplikačných protokolov.
2. Overenie údajov: Môžu sa použiť na overenie formátu vstupných údajov, ako sú e-mailové adresy, telefónne čísla a ID používateľov.
3. Hľadať a nahradiť: Regulárne výrazy možno použiť na nájdenie a nahradenie konkrétnych vzorov v texte alebo kódovej základni.
4. Premenovanie súborov: Môžu sa použiť na dávkové premenovanie súborov na základe určitých vzorov.
5. Analýza vstupu používateľa: Regulárne výrazy možno použiť na analýzu vstupu používateľa alebo príkazových riadkov.
6. Čítanie konfiguračných súborov: Môžu sa použiť na čítanie a analýzu konfiguračných súborov.

2 Ekvivalencia regulárnych výrazov

Regulárny výraz je jazyk na opis množiny reťazcov, ktorým môže zodpovedať, a zvyčajne existuje viac ako jeden spôsob jeho vyjadrenia. Napríklad číslicu možno opísať ako rozsah znakov [0-9] a možno ju opísať aj pomocou skratky d. Znak slova vyjadrený skratkou je ekvivalentný [A-Za-z-0-9]. Niekedy si hlásenia o chybách vyžadujú riešenie pomocou transformácií zachovávajúcich sémantiku, ktoré zlepšujú výkonnosť, a nie úpravami správania regulárnych výrazov. [?] V jednom hlásení o chybe sa všetky zachytávajúce skupiny regulárnych výrazov zmenili na nezachytávajúce skupiny (napr. (r n— r— n— f) na (r n— r— n— f)), aby sa zabránilo spätnému sledovaniu a rozsah regulárneho výrazu sa mutáciou nezmenil. V inom príklade je regulárny výraz (W— d—-) zmenený na [$\wedge R-Za-z$]$kvlilepejitatenostiregulrnehovrazu.(Navysvetlenie,\ Wjeekvivalentnnegovanejtriedeznakov[\wedge R-Za-z-0-9]a\ djeekvivalentn[0-9];take[0-9]jeplatnvpoadovanejtriedeznakov;podobnesazaobchdzaa1001[?]Tieposkytujdkaz, enievetkypravymeniasprvanie, atedanievetkysnahyotestovaniebysamalizan$

3

Pre regulárne výrazy [] existujú nástroje na generovanie testov regulárnych výrazov. Tieto nástroje vymenúvajú členy jazyka regulárnych výrazov, ale nevypočítavajú nezhodujúce sa reťazce. Nedávne výskumné úsilie skúmalo prístup založený na chybách pri generovaní testov regulárnych výrazov prostredníctvom testovania mutácií []; do regulárnych výrazov sa vnášajú chyby a generujú sa reťazce, ktoré sú svedkami chyby, čím sa poskytujú príklady nezhodného správania pôvodného regulárneho výrazu. Hoci je testovanie mutáciou zrelou oblasťou výskumu, na mutáciu regulárnych výrazov sa zamerali nedávno []. Tieto snahy však definovali operátory mutácie ad hoc spôsobom, bez toho, aby zohľadnili, ako sa regulárne výrazy v praxi skutočne vyvíjajú. Preto injektované chyby nemusia byť reprezentatívne pre skutočné úpravy.

4 Záver

Literatúra

[1] Carl Chapman and Kathryn T. Stolee. Exploring regular expression usage and context in python. In *Proceedings of the 25th International Symposium on Software Testing and Analysis*. ACM, July 2016.

[2] Carl Chapman, Peipei Wang, and Kathryn T. Stolee. Exploring regular expression comprehension. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, October 2017.

[3] Louis G. Michael, James Donohue, James C. Davis, Dongyoon Lee, and Francisco Servant. Regexes are hard: Decision-making, difficulties, and risks in programming regular expressions. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, November 2019.

[4] Peipei Wang, Gina R. Bai, and Kathryn T. Stolee. Exploring regular expression evolution. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, February 2019.

[5] Peipei Wang and Kathryn T. Stolee. How well are regular expressions tested in the wild? In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM, October 2018.