

Vizualizace regulárních výrazů

Regular Expression Visualization

Dominik Kundra

Bakalářská práce

Vedoucí práce: Ing. Jakub Beránek

Ostrava, 2024

Zadání bakalářské práce

Student:

Dominik Kundra

Studijní program:

B0613A140014 Informatika

Téma:

Vizualizace regulárních výrazů
Regular Expression Visualization

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je vytvořit nástroj sloužící pro vizualizaci a ladění regulárních výrazů. Nástroj by měl být schopný zpracovat zvolený regulární výraz, sestavit plán vykonávání daného výrazu dle zvolené implementace a poté umožnit programátorovi interaktivně krokovat provádění regulárního výrazu. Nástroj by měl být vytvořen jako rozšíření do vývojového prostředí (např. do Visual Studio Code), aby šel jednoduše použít při vývoji programů. Výsledná aplikace by měla být řádně zdokumentována a při jejím vývoji by měl být využit verzovací systém (např. git).

1. Analyzujte a popište možnosti implementace regulárních výrazů.
2. Navrhnete architekturu rozšíření do vývojového prostředí, které bude schopné analyzovat regulární výrazy ze zvoleného zdrojového kódu.
3. Naimplementujte nástroj pro vizualizaci regulárního výrazu a integrujte jej do vývojového prostředí.
4. Otestujte vizualizaci nástroje na regulárních výrazech z reálných projektů.

Seznam doporučené odborné literatury:

- [1] FRIELD, Jeffrey. Mastering Regular Expressions 3rd Edition. 2006. O'Reilly Media. ISBN: 978-0596528126
- [2] SORVA, Juha. Visual program simulation in introductory programming education. Espoo: Aalto Univ. School of Science, 2012. ISBN 9789526046266. Dostupný také z WWW: <http://doi.acm.org/10.1145/2445196.2445368>.
- [3] VANDERKAM, Dan. Effective TypeScript : 62 Specific Ways to Improve Your TypeScript. O'Reilly Media, 2019. ISBN 978-1492053699

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jakub Beránek**

Datum zadání: 01.09.2023

Datum odevzdání: 30.04.2024

Garant studijního programu: doc. Mgr. Miloš Kudělka, Ph.D.

V IS EDISON zadáno: 09.11.2023 15:43:31

Abstrakt

No Czech or Slovak abstract is given

Klíčová slova

No Czech or Slovak keywords are given

Abstract

No English abstract is given

Keywords

No English keywords are given

Obsah

Seznam obrázků	5
Seznam tabulek	6
1 Úvod	7
Přílohy	7
A Dlouhý zdrojový kód	8

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Vyhledávání v textu patří mezi základní problémy, se kterými se velice pravděpodobně potká skoro každý programátor. Tento problém se dá řešit mnoha způsoby, avšak ne všechna řešení lze použít univerzálně a každý ze způsobů má své výhody a nevýhody. Jedním ze přístupů je využití konečných automatů, kterými se bude zabývat tato práce a to ve formě regulárních výrazů.

Cílem této práce je naimplementovat nástroj, který bude umožňovat procházení regulárních výrazů a následné vizualizování těchto průchodů, jako rozšíření do vývojového prostředí.

Při vývoji programů, se programátor může setkat s regulárními výrazy, jedná se totiž o poměrně rychlé řešení pro vyhledávání v textu. Tyto výrazy se však brzy mohou stát hůře čitelnými, jelikož neumožňují žádné formátování¹. Z tohoto důvodu se hodí mít nástroj, který potencionálně usnadní práci programátorům, aby si mohly zobrazit průchod zadaným výrazem. Sice existující řešení tohoto problému již existují, ale pro zvolené vývojové prostředí mnoho přístupů neexistuje. Tato situace je motivací, zabývat se problémem do hloubky a nabídnout originální řešení v daném směru.

¹upravení vzhledu, tvaru

Příloha A

Dlouhý zdrojový kód

```
#include <climits>
#include "ArraySortingAlgorithms.h"

void Exchange(int& x, int& y)
{
    int aux = x;
    x = y;
    y = aux;
}

void SelectSort(int a[], const int n)
{
    for (int i = 0; i < n; i++)
    {
        int min = i;
        for (int j = i + 1; j < n; j++)
        {
            if (a[j] < a[min])
            {
                min = j;
            }
        }
        Exchange(a[min], a[i]);
    }
}
```



```

void InsertSort(int a[], const int n)
{
    for (int i = 1; i < n; i++)
    {
        int v = a[i];
        int j = i;
        while (j > 0)
        {
            if (a[j - 1] > v)
            {
                a[j] = a[j - 1];
                j -= 1;
            }
            else
            {
                break;
            }
        }
        a[j] = v;
    }
}

```

```

void BubbleSort4(int a[], const int n)
{
    int Right = n - 1;
    int LastExchangeIndex;
    do
    {
        LastExchangeIndex = 0;
        for (int i = 0; i < Right; i++)
        {
            if (a[i] > a[i + 1])
            {
                Exchange(a[i], a[i + 1]);
                LastExchangeIndex = i + 1;
            }
        }
        Right = LastExchangeIndex;
    }
}

```

```

    } while (LastExchangeIndex > 0);
}

void ShakerSort(int a[], const int n)
{
    int ExchangeIndex = 0;
    int Left = 0;
    int Right = n - 1;
    do
    {
        for (int i = Left; i < Right; i++)
        {
            if (a[i] > a[i + 1])
            {
                Exchange(a[i], a[i + 1]);
                ExchangeIndex = i;
            }
        }
        Right = ExchangeIndex;
        for (int i = Right; i > Left; i--)
        {
            if (a[i - 1] > a[i])
            {
                Exchange(a[i - 1], a[i]);
                ExchangeIndex = i;
            }
        }
        Left = ExchangeIndex;
    } while (Left < Right);
}

```

Listing A.1: Dlouhý zdrojový kód v jazyce C++ načtený s externího souboru