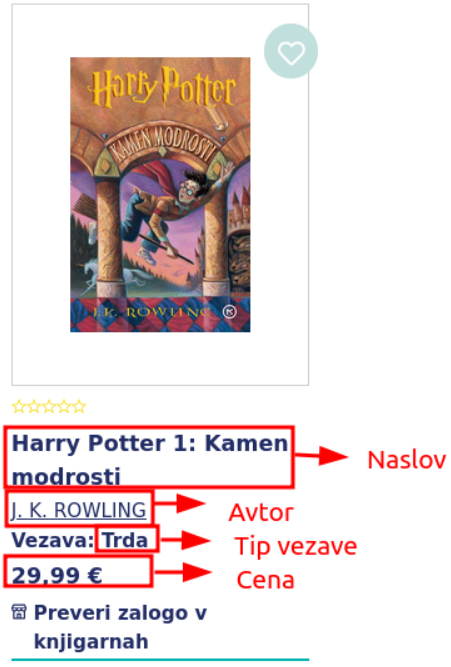
Poročilo 2. domače naloge - ekstrakcija podatkov

May 10, 2024

1 Uvod

Namen 2. domače naloge je bila implementacija ekstrakcije podatkov s treh parov strani na tri načine: z regularnimi izrazi, z XPathom in na podlagi implementacije avtomatskega splošnega algoritma.

Kot tretjo vrsto spletne strani smo si izbrali stran **emka.si**. V iskalnih na spletni strani smo vnesli dva različna iskalna niza, ki sta prinesla rezultate različnih dolžin. Na sliki 1 lahko vidimo podatke, ki smo ji opredelili kot tiste, ki jih želimo ekstrahirati iz seznama ponudbe knjig.



Slika 1: Opredeljeni podatki iz kartic ponudb knjig na spletni strani *emka.si*.

2 Regularni izrazi

2.1 Overstock

Za spletno stran Overstock je bilo potrebno izvleči več artiklov, ki se pojavijo na strani. Ti artikli imajo vsi enako strukturo, torej je potreben le splošni regularni izraz, ki bo pravilno deloval za vse artikle. Vsi podatki artikla so zajeti naenkrat, z uporabo enega samega izraza. Vsak artikel ima tudi svoj unikaten PROD_ID, po katerem med seboj ločimo posamezne artikle. To naredimo z uporabo "backreference", ki v regularen izraz vstavi vnaprej grupiran izraz. To je narejeno z značko `\1` na koncu izraza. Da se zajamejo vsi artikli, izvajamo regularni izraz v zanki, dokler dobivamo ujemanja.

Regularen izraz:

```
href="http://www.overstock.com/cgi-bin/d2.cgi\?PAGE=PROFRAME&amp;
PROD_ID=(\d*)"><b>(.*)</b>[\s\S]*(\$.*)</s>[\s\S]*(\$.*)</b>[\s\S]
*(\$.*) (\(.*\))</span>[\s\S]*"normal">([\s\S]*)<br><a href="http:
//www.overstock.com/cgi-bin/d2.cgi\?PAGE=PROFRAME&amp;PROD_ID=\\1"
```

2.2 RTV

Za spletno stran MMC RTV Slovenija smo uporabili en regularen izraz za vsak zajet element strani, razen za polja *Author* in *PublishedTime*, kjer je en izraz zadostoval za zajem obeh. Ker je vsebina strani enkratna, se vsi regularni izrazi izvedejo le enkrat.

Regularni izrazi:

- Avtor, čas objave:

  ```
  class="author-timestamp">[.\s]*<strong>([\w\s]+)<\/strong>\| ([\w\d\.: ]+)\t
  ```

- Naslov

  ```
  <h1>(.*)<\/
  ```

- Podnaslov

  ```
  class="subtitle">(.*)<\/
  ```

- Uvodnik

  ```
  class="lead">(.*)<\/
  ```

- Jedro

  ```
  <p class="Body"><\/p><p class="Body">(.*)\n
  ```

2.3 Emka

- Naslov

```
class="ie-book-title[^>]*>(.*?)<span
```

- Avtor

```
<li class="ie-custom-grid tw-relative".*?<a[^>]*
    class="tw-text-darkblue tw-text-sm tw-underline"[^>]*>\s*(.*?)\s*\ \</a>
```

- Vezava

```
<div class="product_var tw-text-darkblue tw-text-sm tw-font-bold">
    Vezava:\s*(.*?)\s*</div>
```

- Cena

```
<li class="ie-custom-grid tw-relative".*?<div class="book-item-buy">
    \s*<div[^>]*>\s*<div[^>]*>\s*<span[^>]*>(.*?)</span>
```

3 XPath

V primeru vseh treh strani vsak izraz lahko zajame samo en element strani.

3.1 Overstock

- Naslovi:

```
f'{OVERSTOCK_PREFIX}/a/b/text()'
```

- Seznam cen

```
f'{OVERSTOCK_PREFIX}/table/tbody/tr/td[1]/table/tbody/tr[1]/td[2]/s/text()'
```

- Cene

```
f'{OVERSTOCK_PREFIX}/table/tbody/tr/td[1]/table/tbody/tr[2]/td[2]/span/b/text()'
```

- Skupni prihranek

```
f'{OVERSTOCK_PREFIX}/table/tbody/tr/td[1]/table/tbody/tr[3]/td[2]/span/text()'
```

3.2 RTV

- Avtor: *//div[@class="author"]/div[@class="author-name"]/text()*
- Čas objave: *//div[@class="publish-meta"]/text()*
- Naslov: *//header[@class="article-header"]/h1/text()*
- Podnaslov: *//header[@class="article-header"]/div[@class="subtitle"]/text()*
- Uvodnik: *//header[@class="article-header"]/p[@class="lead"]/text()*
- Jedro: *//div[@class="article-body"]/article/p/text()*

3.3 Emka

- Naslov: *//a[contains(@class, "ie-book-title")]/text()*
- Avtor: *//div[@class="book-item-information tw-relative"]/a[2]/text()*
- Vezava: *//div[@class="book-item-information tw-relative"]/div[3]/text()*
- Cena: *//div[@class="book-item-buy"]/div/div/span/text()*

4 Zaključek

Žal nam implementacija ključnega dela naloge ni uspela.