



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR INFORMATIK
LEHR- UND FORSCHUNGSEINHEIT FÜR
PROGRAMMIER- UND MODELLIERUNGSSPRACHEN



Web-Informationssysteme, WS 2009/10

Seminar-Übung 6: Klausuraufgaben: Sneak Preview

Besprechung am Mi 16.12.2009

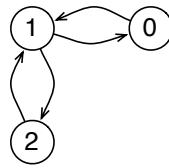
— SÜ-6.1 —

Klausuraufgaben: Sneak Preview

Wiederholung

SÜ-6.1.1 PageRank

Gegeben sei der folgende Web-Graph mit Web-Seiten $\{0, 1, 2\}$ und Links/Kanten wie angegeben:



1. Berechnen Sie für diesen Web-Graphen die Übergangs- (oder Google-) Matrix mit einem *random leap* Faktor von $\alpha = 0.5$!

Hinweis: In Zelle (i, j) der Matrix soll die Wahrscheinlichkeit stehen von Seite j zu Seite i zu surfen.

$$\begin{pmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix}$$

2. Berechnen Sie, ausgehend von einem Startvektor $\vec{x}_0 = (1, 0, 0)$, die ersten drei Schritte der Iteration der Power-Methode zur Berechnung des PageRanks für den obigen Web-Graphen

$$\vec{x}_1 = \begin{pmatrix} \dots & \dots & \dots \end{pmatrix}$$

$$\vec{x}_2 = \begin{pmatrix} \dots & \dots & \dots \end{pmatrix}$$

$$\vec{x}_3 = \begin{pmatrix} \dots & \dots & \dots \end{pmatrix}$$

3. Leiten Sie daraus ab, welcher der drei folgenden Vektoren der PageRank-Vektor zu obigem Web-Graphen ist!

☐ $(5/18, 4/9, 5/18)$

☐ $(1/3, 1/3, 1/3)$

☐ $(7/12, 1/6, 1/4)$

SÜ-6.1.2 HTML & XML

sÜ-6.1.2.1 LÄNDER, PROVINZEN, STÄDTE—BEISPIELDOKUMENT

```

1 <?xml version="1.0" encoding="UTF-8"?>
  <middleearth>
3    <country id="GD" capital="Osgiliath">
      <name>Gondor</name>
5      <population><rural>700000</rural><urban>200000</urban></population>
      <border country="MD" />
7      <province id="ithilien" country="GD">
        <name>Ithilien</name>
9        <population>182000</population>
        <city id="Osgiliath" is_country_cap="yes" country="GD">
11          <name>Osgiliath</name>
          <capitol>Rond Giliath</capitol>
13        </city>
      </province>
15      <province id="anorien" country="GD">
        <name>Anorien</name>
17        <population>82000</population>
        <city id="MinasTirith" country="GD">
19          <name>Minas Tirith</name>
          </city>
21        </province>
        <city id="Pelargir" country="GD">
23          <name>Pelargir</name>
        </city>
25      </country>
      <country id="MD" capital="BaradDur">
27        <name>Mordor</name>
        <population><rural>1000000</rural><urban>100000</urban></population>
29        <border country="GD" />
        <province id="gorgoroth" country="MD">
31          <name>Gorgoroth</name>
          <population>574308</population>
33          <city id="BaradDur" is_country_cap="yes" country="MD">
            <name>Barad-Dur</name>
35            <capitol>Dark Tower</capitol>
          </city>
37          </province>
          <province id="ithilien-md" country="MD">
39            <name>Ithilien</name>
            <population>82000</population>
41            <city id="MinasMorgul" country="MD">
              <name>Minas Morgul</name>
43            </city>
          </province>
45        </country>
      </middleearth>

```

sÜ-6.1.2.2 CSS

In Abschnitt **SÜ-6.1.2.1** ist ein Beispieldokument über Länder, Provinzen und Städte in Mitteleuropa gegeben.

1. Ordnen Sie jedem der folgenden CSS Selektoren eine der rechts-stehenden Elementmengen zu, so dass der Selektor die Elemente dieser Menge selektiert. Elemente sind durch Zeilennummer und Tag referenziert.

country[id='GD'] > city	a) \emptyset
country[id='GD'] city	b) {city ₂₂ }
#MD > *:nth-child(2n+1)	c) {city ₁₀ , city ₁₈ }
#MD *:nth-child(3n-1)	d) {city ₁₀ , city ₁₈ , city ₂₂ }
#GD province > name ~ *[country="GD"]	e) {name ₂₇ , border ₂₉ , province ₃₈ }
province#ithilien name + *[country="GD"]	f) {population ₂₈ , population ₃₂ , capitol ₃₅ , province ₃₈ , population ₄₀ }

2. Ergänzen Sie **ausschließlich** die Selektoren der folgenden beiden CSS Regeln, so dass Städte in Mordor in roter Farbe, aber Städte mit is_country_cap = yes **stets** (sowohl in als auch außerhalb von Mordor) in blauer Farbe erscheinen. Sie dürfen die Reihenfolge der Regeln *nicht* verändern!

Hinweis: Eine Regel kann mehrere, durch , getrennte Selektoren verwenden.

```

.....
2 ..... {
   color: blue; }
4
.....
6 ..... {
   color: red; }

```

3. Für welche Fragmente von CSS gilt: Es gibt für jeden CSS Selektor in diesem Fragment einen XPath-Ausdruck, der dieselben Knoten selektiert?

- ☐ volles CSS3
 ☐ CSS3 ohne Pseudo-Klassen wie :hover
 ☐ CSS3 ohne ~
- ☐ CSS3 ohne Pseudo-Klassen und Pseudo-Elemente
- ☐ CSS3 ohne Pseudo-Klassen, Pseudo-Elemente und Attribut-Selektoren

SÜ-6.1.2.3 DOKUMENTE RATEN

Ausdruck	Ergebnis
<code>count(//*)</code>	10
<code>count(/middleearth/country/following-sibling::river)</code>	1
<code>count(//country)</code>	1
<code>count(//province)</code>	2
<code>count(//country[id='IG']/province)</code>	1
<code>count(//province/province)</code>	1
<code>count(//country/province/*)</code>	2
<code>count(//province[name])</code>	2
<code>count(//name)</code>	3
<code>count(//river[name])</code>	1
<code>count(//province/name[.='Nan Curunir'] /following-sibling::located_at)</code>	1
<code>count(//located_at[@watertype='river' and @ref='isen'])</code>	1
<code>count(//located_in/..[@id='isen'])</code>	1
<code>count(//@*)</code>	9
<code>count(//@country)</code>	2
<code>count(//country/province/@country)</code>	1
<code>count(//@id)</code>	3
<code>count(//country/@id //province/province/@id //river/@id)</code>	3
<code>count(id(//country/@central)/self::province[@id='curunir'])</code>	1
<code>count(//river[@country='IG']/located_in[@province='curunir'])</code>	1
<code>count(id(//@country)</code>	1
<code>//name/text()</code>	{"Westfold", "Nan Curunir", "Isen"}
<code>//@id</code>	{"IG", "curunir", "isen"}

Vervollständigen Sie das unten angegebene Skelett eines XML-Dokument, so dass alle obigen XPath Ausdrücke das angegebene Ergebnis auf dem von Ihnen angegebenen Dokument liefern.

Hinweis: Struktur und Elemente sind nicht unähnlich zu dem Dokument von Abschnitt SÜ-6.1.2.1, aber über eine andere Region Mittelherdes.

```

<middleearth>
2  < ..... >
   .....
4  .....
   .....
6  .....
   .....
8  .....
   .....
10 .....
   </ ..... >
12 < ..... >
   <name> ..... </name>
14 <located_in ..... />
   </ ..... >
16 </middleearth>

```

Welche der folgenden Aussagen zur **Komplexität der Auswertung von XPath** sind korrekt?

Hinweis: Wir betrachten kombiniert (*combined*) Komplexität, d.h. sowohl die Größe der Eingabe als auch die der Query sind variabel.

- ☐ volles XPath lässt sich in polynomieller Zeit und Speicher auswerten
- ☐ es gibt **kein** Fragment von XPath, das sich in linearer Zeit und Speicher auswerten lässt
- ☐ XPath **ohne Prädikate** lässt sich in linearer Zeit und Speicher auswerten
- ☐ XPath lässt sich nur mit **mehr** Zeit und Speicher auswerten als SPARQL