UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA

Milica Knežević

Mere sličnosti XML podataka i primene u otkrivanju duplikata

MASTER RAD

Mentor: dr Jelena Ivetić

Novi Sad, 2015.

УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:		
Идентификациони број, **ИБР**:		
Тип документације, **ТД**:		Монографски рад
Тип записа, **ТЗ**:		Штампа
Врста рада, **ВР**:		Мастер рад
Аутор, **АУ**:		Милица Кнежевић
Ментор, **МН**:		др Јелена Иветић
Наслов рада, **НР**:		Мере сличности XML података и примене у откривању дупликата
Језик публикације, **ЈП**:		Српски
Језик извода, **ЈИ**:		Српски
Земља публиковања, **ЗП**:		Србија
Уже географско подручје, **УГП**:		Војводина
Година, **ГО**:		2015.
Издавач, **ИЗ**:		Ауторски препринт
Место и адреса, **МА**:		Факултет техничких наука, Нови Сад
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)		(7/35/0/0/8/0/0)
Научна област, **НО**:		Математика
Научна дисциплина, **НД**:		Примењена математика
Предметна одредница/Кључне речи, **ПО**:		XML модели, растојање између стабала, откривање дупликата
УДК		
Чува се, **ЧУ**:		У библиотеци Факултета техничких наука
Важна напомена, **ВН**:		
Извод, **ИЗ**:		У овом мастер раду бавимо се проблемом утврђивања сличности XML података у циљу откривања дупликата и дајемо приказ једне мере сличности којом се превазилазе проблеми који се могу јавити када се класична мера растојања између стабала примени на XML стабла.
Датум прихватања теме, **ДП**:		
Датум одбране, **ДО**:		29.12.2015.
Чланови комисије, **КО**:	Председник:	др Биљана Михаиловић
	Члан:	др Срђан Сладојевић — Потпис ментора
	Члан, ментор:	др Јелена Иветић

UNIVERSITY OF NOVI SAD ● **FACULTY OF TECHNICAL SCIENCES**
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	Monographic
Type of record, **TR**:	Printed
Contents code, **CC**:	Master thesis
Author, **AU**:	Milica Knežević
Mentor, **MN**:	Jelena Ivetić, PhD
Title, **TI**:	XML similarity measures with applications in duplicate detection
Language of text, **LT**:	Serbian
Language of abstract, **LA**:	Serbian
Country of publication, **CP**:	Serbia
Locality of publication, **LP**:	Vojvodina
Publication year, **PY**:	2015.
Publisher, **PB**:	Authors reprint
Publication place, **PP**:	Faculty of Technical Sciences, Novi Sad
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	(7/35/0/0/8/0/0)
Scientific field, **SF**:	Mathematics
Scientific discipline, **SD**:	Applied mathematics
Subject/Key words, **S/KW**:	XML models, tree edit distance, duplicate detection
UC	
Holding data, **HD**:	The library of the Faculty of Technical Sciences
Note, **N**:	
Abstract, **AB**:	In this master thesis, we discuss the XML similarity problem in the context of duplicate detection and present a similarity measure which can be used to overcome issues that may appear when the classic tree edit distance measure is applied to XML trees.
Accepted by the Scientific Board on, **ASB**:	
Defended on, **DE**:	December 29th, 2015.
Defended Board, **DB**: President:	Biljana Mihailović, PhD
Member:	Srđan Sladojević, PhD — Menthor's sign
Member, Mentor:	Jelena Ivetić, PhD

	УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА** 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Број:
	ЗАДАТАК ЗА МАСТЕР РАД	Датум:

(Податке уноси предметни наставник - ментор)

СТУДИЈСКИ ПРОГРАМ:	Математика у техници
РУКОВОДИЛАЦ СТУДИЈСКОГ ПРОГРАМА:	Проф. др Јованка Пантовић

Студент:	Милица Кнежевић	Број индекса:	V1 12/2015
Област:	примењена математика		
Ментор:	др Јелена Иветић		

НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА МАСТЕР РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:

- проблем – тема рада;
- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;

НАСЛОВ МАСТЕР РАДА:

Мере сличности XML података и примене у откривању дупликата

ТЕКСТ ЗАДАТКА:

Увести основне појмове XML технологија и представити синтаксу XML језика. Формално описати математички модел за представљање XML података заснован на моделу стабла. Дефинисати мере сличности XML података дате на овом моделу, засноване на растојању између стабала. Дефинисати дупликате код XML података, хомогене и хетерогене, и применити уведене мере сличности на откривање дупликата.

Руководилац студијског програма:	Ментор рада:

Примерак за: O - Студента; O - Ментора

Sadržaj

1 Uvod **2**

2 XML **4**
2.1 Sintaksa i validnost 5
2.2 Vrste XML dokumenata 8
2.2.1 Tekst-centričan XML 8
2.2.2 Podatak-centričan XML 9

3 Model stabla **11**

4 Mere sličnosti **15**

5 Otkrivanje duplikata u XML podacima **18**
5.1 Homogeni i heterogeni duplikati 20
5.2 Primer algoritma za otkrivanje XML duplikata 23

6 Zaključak **33**

7 Literatura **34**

1 Uvod

Pri integraciji podataka iz više različitih izvora gotovo neizostavno se javlja problem višestrukih pojavljivanja zapisa koji odgovaraju istom objektu. Otkrivanje i povezivanje ovakvih zapisa često je deo procesa integracije skupova podataka koji ne dele zajednički identifikator (primarni ključ objekta u bazi podataka, URI, JMBG osobe, ISBN knjige, i sl.) koji bi ih nedvosmisleno povezao sa objektom iz „stvarnog sveta" koji opisuju. Ovaj problem prvi put pominje Dunn, sredinom XX veka, u radu pod nazivom „Record Linkage"[8] objavljenom u časopisu „American Journal of Public Health". Teorijske osnove problema 1959. godine postavlja Newcombe, a formalni model uvode Fellegi i Sunter 1969. Problem se, ironično, pojavljuje pod više različitih naziva, a to su, pored izvornog termina *record linkage*, još i: *object identification*, *data matching*, *coreference resolution*, *entity resolution*, *record resolution*, *record matching*, *fuzzy matching*, *approximate matching*, *merge/purge*, itd.. U ovom radu koristićemo termin **otkrivanje duplikata** (eng. *duplicate detection*).

Problem otkrivanja duplikata dobro je poznat i intenzivno proučavan u oblasti relacionih baza podataka, a sa nastankom XML-a i ogromnim količinama podataka koji se čuvaju i razmenjuju u ovom formatu, otkrivanje XML duplikata postaje popularna tema za istraživanje. Većinom se pristupi za rešavanje problema duplikata u XML podacima zasnivaju na otkrivanju sličnih zapisa pomoću prikladno odabrane mere i praga sličnosti. Model stabla je jedan od najpoznatijih i najčešće korišćenih modela za predstavljanje XML podataka. U slučaju ovakve reprezentacije, prirodan izbor pri utvrđivanju sličnosti predstavljaju mere rastojanja između stabala (eng. *tree edit distance*). Ove mere pretpostavljaju totalno uređenje na skupu čvorova u stablu. Međutim, takvo ograničenje je, kada je reč o XML podacima, previše strogo jer redosled elemenata koji imaju istog roditelja često nije od značaja. Kao posledica direktne primene ovih mera mogla bi se javiti velika količina duplikata koji nisu prepoznati kao takvi, sto bi značilo loš kvalitet dobijenih rezultata. Osim toga, uočeno je da neke od standardnih operacija koje se koriste pri utvrđivanju rastojanja između stabala, u kontekstu XML podataka mogu dovesti do narušavanja semantike i pogrešne interpretacije XML sadržaja. U ovom master radu dajemo prikaz jednog pristupa otkrivanju duplikata u XML podacima predloženog u [15], koji uzima u obzir specifičnosti XML stabala i nastoji da otkloni uočene probleme.

Ostatak rada organizovan je na sledeći način. U Poglavlju 2 uvodimo osnovne pojmove XML tehnologija, uz kratak osvrt na istoriju nastanka XML formata i njegove prednosti koje su razlog njegove opšte prihvaćenosti kao standarda za opisivanje, čuvanje i razmenu podataka. Prikazane su osnove sintakse jezika XML i pojmovi dobro formiranog (eng. *well-formed*) i validnog XML dokumenta.

U Poglavlju 3 bavimo se poznatim modelima za predstavljanje XML dokumenata i dajemo detaljan prikaz modela korenskog, označenog stabla, kao jednog od najviše korišćenih.

O utvrđivanju sličnosti pomoću mere rastojanja između stabala (eng. *tree edit distance*) biće reči u Poglavlju 4.

Problemom otkrivanja duplikata bavimo se u Poglavlju 5. Prvo dajemo generalni prikaz postupka za otkrivanje duplikata koji se zasniva na utvrđivanju sličnosti između različitih zapisa pomoću odabrane mere i zadatog praga sličnosti. Zatim se upoznajemo sa dva tipa duplikata, homogenim i heterogenim. Kao homogeni duplikati kategorizovani su oni XML dokumenti koji su u skladu sa istom XML shemom, odnosno DTD, a u suprotnom je reč o heterogenim duplikatima. Potom dajemo detaljan opis mere sličnosti za XML podatke i rešenja za otkrivanja XML duplikata zasnovanog na ovoj meri, koji su predloženi u [15].

U Poglavlju 6 sumirana je predstavljena tema i dati zaključni komentari.

2 XML

Jezici za obeležavanje (eng. *markup languages*) omogućavaju da se tekstualnom dokumentu dodaju metapodaci, informacije o tekstu i njegovoj strukturi. Ove informacije su uključene u sam tekst i to na takav način da se mogu identifikovati i sintaksno razlikovati od polaznog sadržaja. XML (eng. *Extensible Markup Language*) je metajezik nastao u drugoj polovini 1990-tih. Dizajniran je za potrebe opisa, prenosa i čuvanja podataka. Nastao je kao derivat SGML-a (eng. *Standard Generalized Markup Language*)[11]. XML je pravljen kao podskup SGML-a čije bi procesiranje, u odnosu na procesiranje SGML-a, bilo olakšano i pojednostavljeno, dok bi se u velikoj meri zadržala izvorna fleksibilnost[1]. XML je objavljen kao preporuka W3 konzorcijuma (eng. *World Wide Web Consortium*, `http://www.w3.org/`) u februaru 1998. godine.

Kako je XML metajezik, ne postoji predefinisan skup dozvoljenih reči kojima se opisuje struktura podataka u dokumentima, već sam autor, u zavisnosti od potrebe i konkretne primene, određuje ove osnovne gradivne jedinice, pridržavajući se pri tom sintaksnih pravila koja definišu opštu formu dobro formiranog XML dokumenta, a o kojima će biti više reči u Poglavlju 2.1.

Kao glavne prednosti XML formata, mogu se izdvojiti sledeće:

- Robustan je, fleksibilan i omogućava opisivanje vrlo kompleksnih struktura podataka;
- Omogućava prenos podataka koji je nezavisan od hardverske i softverske platforme;
- Postoji veliki broj gotovih rešenja (alata i biblioteka) za rad sa XML-om, što značajno olakšava manipulisanje XML dokumentima, kao i razvoj softvera koji koristi XML format za opis podataka;
- Slobodan je (eng. *non-proprietary*), perzistentan format za koji postoje mehanizmi verifikacije;
- Dizajniran je da bude čitljiv kako računarima, tako i ljudima, što autorima i korisnicima značajno olakšava rad.

[1] Za detaljno poređenje jezika SGML i XML videti: `http://www.w3.org/TR/NOTE-sgml-xml-971215/`

Napomenimo da, sa druge strane, velika fleksibilnost i izražajna moć XML-a utiču na performanse i kompleksnost obrade. Pronalaženje efikasnih metoda i rešenja za pojedine zadatke i probleme, može biti složenije u odnosu na neke druge, tradicionalne formate, o čemu će više reči biti u Poglavlju 5.

2.1 Sintaksa i validnost

Osnovne gradivne jedinice XML dokumenata su elementi i atributi. Svaki XML dokument je hijerarhijski organizovan i predstavlja drvoidnu strukturu sa jednim korenskim elementom. Pri definisanju elemenata i atributa za konkretnu namenu, moraju biti ispoštovana sintaksna pravila koja definišu opštu formu dobro formiranog (eng. *well-formed*) XML dokumenta. Sintaksna pravila su precizno definisana i stroga, ali u isto vreme i vrlo jednostavna. U nastavku ovog poglavlja prikazaćemo osnovna pravila XML standarda koja se odnose na XML elemente i atribute.

Svaki element sastoji se od:

- Etikete za početak (eng. *start-tag*), koja se navodi u formi `<naziv-elementa>`,
- Sadržaja elementa, i
- Etikete za kraj (eng. *end-tag*), u formi `</naziv-elementa>`.

Naziv korenskog elementa mora biti jedinstven na nivou dokumenta, dok se ostali elementi mogu ponavljati. Pri zadavanju naziva elementa moraju biti ispoštovana sledeća pravila:

- Naziv elementa može da počinje malim ili velikim karakterom iz Unicode skupa karaktera, ili donjom crtom;
- Ostali karakteri u nazivu mogu biti još i cifre, srednja crta i tačka;
- Potrebno je voditi računa o malim i velikim slovima, tj. nazivi su case-sensitive;
- Nazivi koji počinju tekstom „XML”, bilo malim ili velikim slovima, su rezervisani i ne bi ih trebalo koristiti (mada u praksi mnogi parseri dopuštaju njihovu upotrebu).

Kao sadržaj elementa mogu se navesti tekst i drugi elementi. Postoje rezervisani karakteri koji se u tekstualnom sadržaju ne smeju navoditi direktno, već pomoću odgovarajuće sekvence. To su karakteri: ampersand `&`, manje `<`, veće `>`, navodnici ″ i apostrof ′. Oni se navode sledećim sekvencama `&`, `<`, `>`, `"` i `'`.

Primer 1. XML dokument sa korenskim elementom `collection` koji sadrži tri dete-elementa: `title`, `provenance` i `subjects`. Element `title` ima tekstualni sardžaj, element `provenance` je prazan[2], a element `subjects` sadrži četiri dete-elementa čiji je sadržaj tekstualan.

[2]Prazan element `provenance` može se navesti i kao `<provenance></provenance>`

```
<collection>
   <title>Books from the First World War period preserved
      by the National Library of Serbia
   </title>
   <provenance/>
   <subjects>
      <subject>World War, 1914-1918</subject>
      <subject>Memoir</subject>
      <subject>Fiction</subject>
      <subject>Military History</subject>
   </subjects>
</collection>
```

XML elementi moraju biti pravilno ugnježdeni. Pod tim se podrazumeva da se etiketa za kraj dete-elementa mora navesti pre etikete za kraj roditelja-elementa.

Primer 2. Nepravilno ugnježdeni XML elementi:

```
<b><i>Nepravilno ugnježdeni elementi</b></i>
```

Atributi predstavljaju parove naziv-vrednost koji se dodeljuju elementima.

Primer 3. Element `title` sa atributima `lang` i `encodinganalog`.

```
<title lang="eng" encodinganalog="3.2.1">
  Books from the First World War period
  preserved by the National Library of Serbia
</title>
```

Za nazive atributa važe ista pravila kao i za nazive elemenata. Dodatno, naziv atributa mora biti jedinstven na nivou elementa. Vrednost atributa može se navesti ili između navodnika ili između apostrofa. Karakteri koji nisu dozvoljeni u vrednostima atributa i moraju se uneti kao sekvenca su manje `<` i ampersand `&`, i umesto njih se navodi `<`, odnosno `&`.

Osim sintaksne ispravnosti, postoje i mehanizmi za definisanje i proveru validnosti strukture dokumenta (npr. svakom elementu koji opisuje knjigu, mora biti pridružen jedinstveni registarski broj i naslov, ali je element koji odgovara sažetku knjige opcioni). XML dokument je validan ako je dobro formiran i ako je u skladu sa pravilima navedenim kroz DTD (Document Type Definition) ili XML shemu (XSD) kojima se preciznije definišu dozvoljeni elementi i atributi. Na ovaj način, moguće je opisati strukturu dokumenta, definisati dozvoljene elemente i atribute, zadati obavezni redosled, broj pojavljivanja elemenata i sl.

Primer 4. Fragment XML dokumenta

```
<metadata>
  <identifier>DE-611-HS-10</identifier>
  <identifier>A: Mann</identifier>
```

```
  <title lang="ger">Brief von Thomas Mann an Oskar Garvens</title>
  <creator>
    <name>Mann, Thomas</name>
  </creator>
  <subject>Garvens, Oskar</subject>
</metadata>
```

može se opisati sledećim DTD-om

```
<!ELEMENT metadata (identifier+, title, creator, subject)>
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT creator (name)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ATTLIST title lang CDATA>
```

dok bi odgovarajuća XML shema bila

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="metadata" type="xs:complexType">
      <xs:sequence>
         <xs:element name="identifier" type="xs:string"
            maxOccurs="unbounded"/>
         <xs:element name="title" type="xs:string">
            <xs:attribute name="lang" type="xs:string"/>
         </xs:element>
         <xs:element name="creator" type="xs:complexType">
            <xs:sequence>
               <xs:element name="name" type="xs:string"/>
            </xs:sequence>
          </xs:element>
          <xs:element name="subject" type="xs:string"/>
      </xs:sequence>
   </xs:element>
</xs:schema>
```

XML shema je noviji format, i za razliku od DTD sintakse, njena sintaksa se zasniva na XML-u. Neke od glavnih razlika izmedju XML sheme i DTD ogledaju se u sledećem:

- Korenski element se ne može definisati pomoću DTD, dok je u XML shemi to moguće uraditi;
- Broj pojavljivanja nekog XML elementa se na jednostavniji i elegantniji način može ograničiti pomoću sheme;
- U XML shemi postoje tipovi podataka, dok DTD ne pruža mogućnost da se definiše tip vrednosti elemenata i atributa;
- Za razliku od DTD, XML shema pruža podršku za imenske prostore.

2.2 Vrste XML dokumenata

Primene XML tehnologija mogu se, u najširem smislu, podeliti na dve velike oblasti, tekst-centrične i podatak-centrične, koje se u osnovi razlikuju po tome da li u dokumentu dominira upravo tekstualni sadržaj ili je struktura podataka stroga i relevantna koliko i sam sadržaj.

2.2.1 Tekst-centričan XML

Tekst-centričan XML naziva se još i dokument-centričan. Kako je nastao iz SGML-a, XML je vrlo brzo prihvaćen za predstavljanje semi-strukturiranih dokumenata kao što su tehnička uputstva, katalozi proizvoda i pravna dokumenta. Glavni sadržaj ovakvih XML dokumenata je semi-strukturirani, obeleženi tekst. Ovi dokumenti su obično namenjeni čoveku kao korisniku, s tim što pre prikaza mogu biti procesirani od strane računara.

Primer 5. Tekst-centrični XML dokument kojim je opisana jedna kolekcija knjiga iz Prvog svetskog rata koja se čuva u Narodnoj biblioteci.

```
<collection> The archival collection
   <title>Books from the First World War period preserved
      by the National Library of Serbia
   </title> has been digitized within the context of the
   <link url="http://www.europeana-collections-1914-1918.eu/">
    Europeana collections 1914-1918
   </link>.
   The collection is <b>freely available online</b> from
   several online catalogues:
   <link url="http://www.vbs.rs/scripts/cobiss?ukaz=getid&amp;lani=en">
      The Library online catalogue
   </link>,
   <link url="http://www.theeuropeanlibrary.org/tel4/collection/a0565">
      The European Library
   </link> and
   <link url="http://www.europeana-collections-1914-1918.eu/">
    The Europeana Collections 1914-1918
   </link>.
   <p>This book collection includes issues from the period of
    <i>World War One</i>, but also from the <i>postwar period</i>,
    if their topic is war-related to the Serbian people.
      The publications included are: <i>literary works</i>
      created during the war, <i>propaganda brochures</i>
      made with the purpose of strengthening the morale
      of the army and the civilians, <i>memoir works</i>
      that evoke the time of great suffering, as well as
      the <i>legislative activity</i> of the Government of
      the Kingdom of Serbia on the island of Corfu. </p>
   <p>Most books are in <i>Serbian</i> but there are
```

```
    <i>English</i>, <i>German</i>, <i>French</i> and
    <i>Italian</i> works, as well.</p>
</collection>
```

Ovakvi XML dokumenti karakterišu se krupno-zrnastim podacima (najmanja nezavisna jedinica može biti i sam dokument). Redosled elemenata je najčešće značajan za razumevanje sadržaja dokumenta, a sam sadržaj je mešovit, tj. elementi sadrže i tekst i druge elemente. Tekst-centričan XML pogodan je za podatke koji su u osnovi tekstualni dokumenti, a potrebno ih je obeležiti kako bi se predstavila osnovna struktura sadržaja.

2.2.2 Podatak-centričan XML

Za razliku od tekst-centričnog, podatak-centrični XML se koristi za strogo strukturirane podatke kao što je, recimo, tekstualna prezentacija sadržaja relacione baze podataka. Podatak-centrični XML dokumenti su najčešće namenjeni za obradu od strane računara, a ne da budu korišćeni direktno od strane čoveka. Obično se upotrebljavaju za razmenu podataka u strukturiranoj formi, kao što je slučaj kod klasičnog EDI (eng. *Electronic Data Interchange*) ili kod aplikacija za rad sa bazama podataka.

Primer 6. Opis kolekcije knjiga iz Prvog svetskog rata koja se čuva u Narodnoj biblioteci pomoću podatak-centričnog XML dokumenta.

```
<collection>
   <title lang="en">Books from the First World War period preserved
      by the National Library of Serbia
   </title>
   <title lang="sr">Kolekcija knjiga iz Prvog svetskog rata
      koja se nalazi u Narodnoj biblioteci Srbije
   </title>
   <provenance>
      <publication>
         <publisher>The European Library</publisher>
         <date>2013-11-11</date>
      </publication>
      <dataprovider>National Library of Serbia</dataprovider>
      <creator>National Library of Serbia</creator>
   </provenance>
   <subjects>
      <subject>World War, 1914-1918</subject>
      <subject>Memoir</subject>
      <subject>Fiction</subject>
      <subject>Military History</subject>
   </subjects>
   <languages>
      <language>Serbian</language>
      <language>English</language>
      <language>German</language>
```

```
        <language>French</language>
        <language>Italian</language>
    </languages>
</collection>
```

Podatak-centrični dokumenti su karakterisani fino-zrnastim podacima (najmanje nezavisne jedinice podataka su na nivou elemenata i atributa) i redosled elemenata često nije od značaja. Dok kod tekst-centričnih XML dokumenata tekstualni sadržaj dominira u odnosu na XML elemente i atribute, kod podatak-centričnih ovaj odnos je znatno uravnoteženiji.

3 Model stabla

XML podaci mogu se predstaviti pomoću nekoliko standardnih, opštih modela podataka, kao što su: model stabla, model usmerenog grafa i vektorski model. U slučaju XML podataka, model bi trebalo da omogući reprezentaciju i sadržaja i karakteristika strukture. Izabrani model predstavlja osnovu za identifikaciju osobina koje se koriste pri utvrđivanju sličnosti XML podataka, kao što ćemo videti u narednim poglavljima. U nastavku ovog poglavlja biće prikazan model označenog, korenskog stabla kao jedan od najčešće korišćenih.

Definicija 1. Neka je V neprazan skup i $E \subseteq \{\{u, v\}|u, v \in V\}$. Uređeni par $G = (V, E)$ naziva se **graf**. Elementi skupa V su **čvorovi** grafa, a elementi skupa E **ivice** ili **grane**.

Definicija 2. Neka je V neprazan skup i $E \subseteq \{(u, v)|u, v \in V\}$. Uređeni par $G = (V, E)$ naziva se **usmereni** ili **orijentisani graf**. Elementi skupa V su **čvorovi** grafa, a elementi skupa E **ivice** ili **grane**.

Definicija 3. **Put** dužine $k(k \geq 1)$ grafa $G = (V, E)$ je niz ivica oblika

$$\{u_0, u_1\}, \{u_1, u_2\}, ..., \{u_{k-1}, u_k\}$$

kod neusmerenog grafa, odnosno

$$(u_0, u_1), (u_1, u_2), ..., (u_{k-1}, u_k)$$

kod usmerenog grafa. Za ovaj put kažemo da počinje u u_0 i završava se u u_k. Čvorovi u_0 i u_k su krajnji čvorovi tog puta.

Definicija 4. **Elementarni** ili **prost put** put je put koji kroz svaki svoj čvor prolazi tačno jednom.

Definicija 5. **Zatvoren** ili **kružni put** je put koji počinje i završava se u istom čvoru.

Definicija 6. **Ciklus** ili **kontura** je elementarni kružni put.

Definicija 7. Za čvorove u i v kažemo da su **povezani** ako postoji put koji počinje u u i završava se u v.

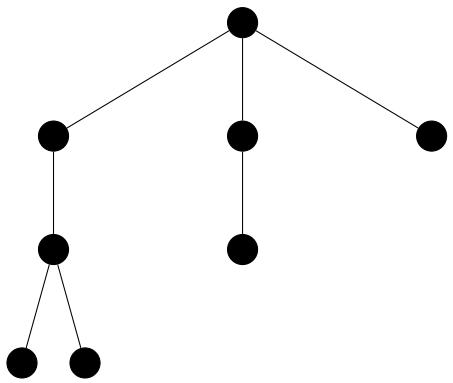
Definicija 8. Graf je **povezan** ako su svaka dva njegova čvora povezana. U suprotnom je graf **nepovezan**.

Definicija 9. **Stablo** je povezan graf bez ciklusa. **Šuma** je disjunktna unija stabala.

Definicija 10. **Korensko stablo** je stablo sa jednim posebno izdvojenim čvorom koji se naziva **koren** stabla, pri čemu su sve ivice implicitno usmerene od korena.

Primer korenskog stabla dat je na Slici 1. Napomenimo da se u slučaju stabala za skup čvorova umesto oznake V često koristi oznaka N (od eng. *nodes*). Ovu konvenciju koristimo i u ostatku rada i stabla označavamo kao uređene parove $T = (N, E)$. U nastavku navodimo nekoliko često korišćenih pojmova u radu sa stablima. Ako za čvorove n_i, n_j korenskog stabla $T = (N, E)$ postoji put $(n_i, n_{i+1}), ..., (n_{j-1}, n_j)$, tada kažemo da je čvor n_i **predak** čvora n_j i da je n_j **potomak** čvora n_i.

Ako je $(n_i, n_j) \in E$, tada kažemo da je čvor n_i **roditelj** ili **direktan predak** čvora n_j i da je n_j **dete** ili **direktan potomak** čvora n_i. Čvor stabla koji nema dece naziva se **list**, dok se ostali čvorovi nazivaju **unutrašnjim**.

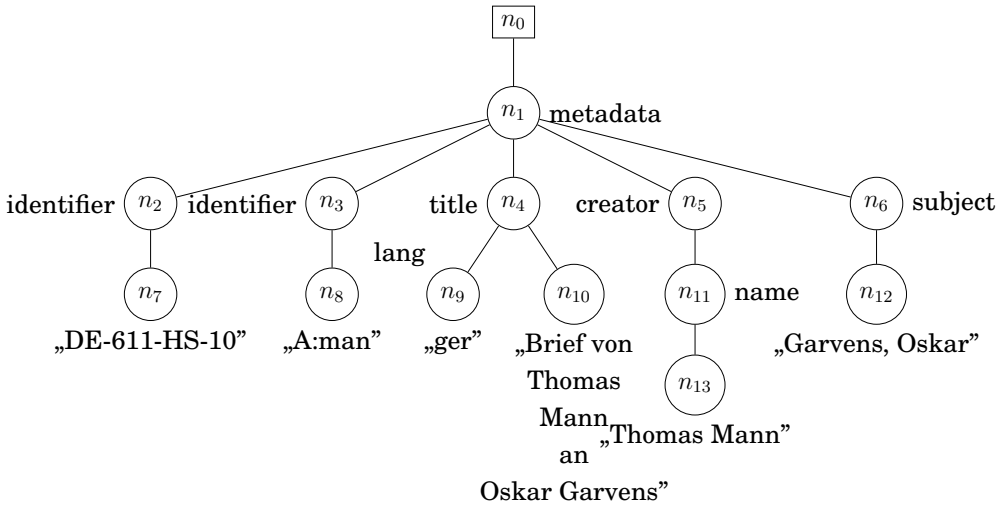


Slika 1: Primer korenskog stabla.

Definicija 11. Za stablo $T = (N, E)$ kažemo da je **uređeno** ako je na skupu čvorova N data relacija totalnog uređenja $<$ takva da za svaki čvor n i svakog njegovog potomka m važi $n < m$. U suprotnom je stablo **neuređeno**.

Definicija 12. Za stablo kažemo je **označeno** ako je svakom čvoru pridružena **oznaka** ili **labela**, pri čemu to pridruživanje ne mora biti injektivno.

Svaki XML dokument može se predstaviti korenskim, označenim stablom čiji čvorovi odgovaraju XML elementima, atributima i tekstualnim vrednostima. Neka je Σ konačan skup oznaka (labela), i neka je D beskonačan skup vrednosti (konstanti). XML elementi i atributi, tj. njihovi nazivi, predstavljaju se pomoću labela. Restrikcija koja se odnosi na kardinalnost skupa oznaka Σ i koja kaže da je ovaj skup konačan neophodna je pri uvođenju nekih logičkih formalizama (videti npr. [13]), a takođe je i u skladu sa praksom, jer se skup dozvoljenih elemenata i atributa obično ograničava pri konkretnoj primeni (pomoću XML sheme ili DTD, kao što smo videli u Poglavlju 2.1). Nasuprot



Slika 2: Primer XML stabla.

tome, skup vrednosti D je beskonačan i njegovi elementi su sve moguće vrednosti XML elemenata i atributa koje se mogu javiti (niske, numeričke vrednosti, datumi itd.).

Definicija 13. XML stablo $T = (N, E, <, \lambda, v)$, čine:

- Korensko stablo (N, E),
- $<$ (parcijalno) uređenje na skupu čvorova N, pri čemu za svaki čvor n_i i svakog njegovog potomka n_j važi da je $n_i < n_j$,
- Parcijalna fukcija $\lambda : N \to \Sigma$ koja čvorovima iz N pridružuje labele iz skupa Σ,
- Parcijalna funkcija $v : N \to D$ koja čvorovima pridružuje vrednosti iz skupa D.

Na Slici 2 prikazana je reprezentacija XML dokumenta iz Primera 4 u formi stabla. Čvor n_0 je dokument čvor, n_1 predstavlja korenski element XML dokumenta, n_2, n_3, n_4, n_5, n_6 i n_{11} predstavljaju odgovarajuće XML elemente, dok je n_9 atribut čvor. Svakom od ovih čvorova pridružena je oznaka koja odgovara nazivu XML elementa, odnosno atributa koji čvor u stablu reprezentuje. Čvorovima $n_7, n_8, n_9, n_{10}, n_{12}$ i n_{13} dodeljene su vrednosti koje odgovaraju tekstualnom sadržaju XML elemenata i atributa.

Definicija 13 modela stabla za XML podatke preuzeta je iz [13], a postoji i značajan broj ranijih verzija modela koje se zasnivaju na stablu, od kojih su neke: izvorna verzija [5], kao i [4][6]. Osim toga, u literaturi se mogu naći modeli koji u izvesnoj meri odstupaju od definicije 13, pa se tako u [12] atributi predstavljaju na isti način kao i elementi, tj. umesto da im se dodeljuje i labela koja odgovara nazivu atributa i vrednost koja odgovara vrednosti atributa, atribut se predstavlja pomoću dva čvora. Prvom čvoru se dodeljuje odgovarajuća labela i taj čvor ima dete (list stabla) kome je dodeljena vrednost atributa. Napomenimo još da po W3C specifikaciji XML standarda, atribut čvorovi koji imaju istog roditelja nisu međusobno uređeni.

Za svako XML stablo T važi sledeće:

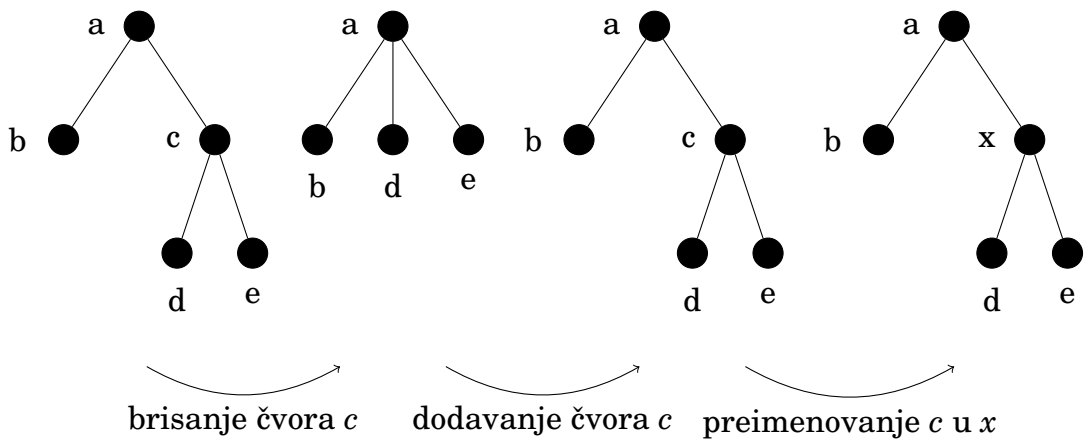
- Skup čvorova N može se predstaviti kao unija disjunktnih podskupova $\{n_0\}, Elem, Attr$ i Txt gde je:
  - n_0 koren stabla T, odnosno dokument čvor,
  - $Elem$ skup element čvorova,
  - $Attr$ skup atribut čvorova,
  - Txt skup tekstualnih čvorova;
- U skupu $Attr \cup Txt$ mogu se naći isključivo listovi stabla; ako $(n_i, n_j) \in E$, tada je $n_i \in \{n_0\} \cup Elem$;
- Za svaki čvor n_i važi da deca koja predstavljaju atribute prethode deci koja odgovaraju elementima i tekstualnom sadržaju, što se formalno može predstaviti na sledeći način, ako $(n_i, n_j) \in E$, $(n_i, n_k) \in E$, $n_j \in Attr$ i $n_k \in Elem \cup Txt$, tada $n_j < n_k$;
- Domen funkcije $\lambda : N \rightarrow \Sigma$ je $Elem \cup Attr$;
- Domen funkcije $v : N \rightarrow D$ je $Attr \cup Txt$;
- Deca dokument čvora n_0 moraju biti element čvorovi, tj. ako $(n_0, n_i) \in E$, tada $n_i \in Elem$.

Model stabla, iako najprirodniji model za predstavljanje XML dokumenata uvodi i neka ograničenja po pitanju predstavljanja relacija koje mogu postojati između XML elemenata. Tako, na primer, XML element može sadržati referencu, tj. link, na drugi element. Referenciranja između elemenata mogu predstavljati važne semantičke informacije, ali se ne mogu prikazati u stablu, pa zahtevaju drugačiji model, kao što je usmereni graf.

4 Mere sličnosti

Neka je T uređeno, označeno stablo. Operacije brisanja, dodavanja i preimenovanja čvorova u T uvodimo na sledeći način:

- **Brisanje** čvora n vrši se tako što se prvo ukloni ovaj čvor kao i sve ivice koje polaze iz njega, a zatim se deca ovog čvora povežu direktno sa njegovim roditeljem;
- **Dodavanje** čvora n kao deteta čvora m na poziciji k vrši se tako što se sva deca čvora m na pozicijama $k, k+1, ...$ uklone i umesto njih se navede novi čvor n, a zatim se uklonjeni čvorovi dodaju kao deca čvora n;
- **Preimenovanje** čvora predstavlja zamenu postojeće oznake datog čvora novom oznakom.



Slika 3: Primer operacija brisanja, dodavanja i preimenovanja čvorova.

Označimo sa ϵ prazan čvor (eng. *null node*). Prethodno navedene operacije nad čvorovima formalno ćemo označavati na sledeći način:

- $n \to \epsilon$ - brisanje čvora n,
- $\epsilon \to n$ - dodavanje čvora n,
- $n \to m$ - preimenovanje, tj. zamena labele čvora n labelom čvora m, gde je $n, m \neq \epsilon$.

Svakoj operaciji dodelimo nenegativan broj koji ćemo nazivati **cenom** operacije i označavati sa γ, uz ograničenje da važi:

(i) $\gamma(n_1 \to n_2) = 0$ ako i samo ako su labele čvorova n_1 i n_2 jednake,

(ii) $\gamma(n_1 \to n_2) = \gamma(n_2 \to n_1)$ - simetričnost,

(iii) $\gamma(n_1 \to n_2) + \gamma(n_2 \to n_3) \geq \gamma(n_1 \to n_3)$ - nejednakost trougla.

Cenu niza operacija $S = s_1, ..., s_k$ određujemo kao:

$$\gamma(S) = \sum_{i=1}^{k} \gamma(s_i).$$

Definicija 14. Rastojanje između stabala (eng. *tree edit distance*) predstavlja minimalnu cenu niza operacija (brisanja, dodavanja, preimenovanja čvorova) kojima se jedno stablo može transformisati u drugo.

Rekurzivno rešenje za utvrđivanje rastojanja između stabala F i G zasniva se na dekompoziciji stabala na podstabla i podšume. U svakom rekurzivnom koraku, dekompozicija se vrši brisanjem ili krajnjeg levog ili krajnjeg desnog korenskog čvora. Pre nego što formalno opišemo algoritam, uvedimo sledeće oznake:

- Neka su n i m oba krajnje levi ili krajnje desni korenski čvorovi u F, odnosno G;
- Sa $F - n$ označavaćemo šumu koja se dobija brisanjem čvora n iz F (i analogno za $G - m$);
- Sa $F - F(n)$ označavaćemo šumu koja se dobija uklanjanjem stabla sa korenom u n iz F (i analogno za $G - G(m)$).

Algoritam za izračunavanje rastojanja δ između F i G dat je na sledeći način:

$$\delta(\emptyset, \emptyset) = 0$$
$$\delta(F, \emptyset) = \delta(F - n, \emptyset) + \gamma(n \to \epsilon)$$
$$\delta(\emptyset, G) = \delta(\emptyset, G - m) + \gamma(\epsilon \to m)$$

ukoliko su F i G stabla:

$$\delta(F, G) = min \begin{cases} \delta(F - n, G) + \gamma(n \to \epsilon) & (1) \\ \delta(F, G - m) + \gamma(\epsilon \to m) & (2) \\ \delta(F - n, G - m) + \gamma(n \to m) & (3) \end{cases}$$

ukoliko je F ili G šuma:

$$\delta(F, G) = min \begin{cases} \delta(F - n, G) + \lambda(n \to \epsilon) & (1) \\ \delta(F, G - m) + \lambda(\epsilon \to m) & (2) \\ \delta(F - F(n), G - G(m)) + \delta(F(n), G(m)) & (4, 5) \end{cases}$$

Rastojanje između F i G računa se rešavanjem tri ili četiri potproblema u zavisnosti od toga da li su i F i G stabla ili ne. Ti potproblemi su: (1) $\delta(F-n, G)$, (2) $\delta(F, G-m)$, (3) $\delta(F-n, G-m)$, (4) $\delta(F-F(n), G-G(m))$ i (5) $\delta(F(n), G(m))$. Potproblemi koji nastaju u svakom koraku dekompozicijom od F i G nazivaju se *relevantni potproblemi* i njihov broj zavisi od načina dekompozicije, tj. od toga da li se n i m izaberu kao krajnje levi ili krajnje desni korenski čvorovi.

Direktnom implementacijom prethodno datog rekurzivnog algoritma, dobija se rešenje koje ima eksponencijalnu složenost. Rešenja kojima se postiže polinomijalna složenost zasnivaju se na paradigmi dinamičkog programiranja, gde se vrednosti dobijene za relevantne potprobleme čuvaju i koriste pri u rešavanju natproblema. Prvo neeksponencijalno rešenje *tree edit distance* problema dao je Tai [19]. Ovo rešenje ima vremensku složenost $O(a^3b^3)$, gde a i b označavaju broj čvorova stabala čije se rastojanje računa. Zhang i Shasha [21] predlažu poboljšanje koje ima složenost $O(a^2b^2)$.

Treba napomenuti da ovo nisu rešenja koja su namenski pravljena za rešavanje problema sličnosti XML podataka, već su nastala u širem kontekstu koji se odnosi na hijerarhijske strukture i zato mogu dati rezulate (nizove operacija brisanja, dodavanja i preimenovanja) koji nisu u potpunosti prikladni kada se primene u XML kontekstu. Tako, primećeno je da operacija brisanja i dodavanja unutrašnjih čvorova mogu narušiti smisao, tj. semantiku sadržanu u XML hijerarhiji. Rešenja koja se zasnivaju na prikazanoj meri rastojanja između stabala, a koja su namenjena specifično za određivanje sličnosti XML podataka, nastoje da zabrane pomenute operacija na unutrašnjim čvorovima i primenjuju ih isključivo na listove [7]. Osim toga, u nekim pristupima predlaže se dopuštanje brisanja unutrašnjih čvorova, ali na takav način da se obriše čitavo podstablo sa korenom u posmatranom čvoru. Jedan takav pristup utvrđivanju sličnosti između XML dokumenata biće dat u narednom poglavlju.

5 Otkrivanje duplikata u XML podacima

Otkrivanje duplikata predstavlja problem identifikacije višestrukih reprezentacija istog objekta iz „stvarnog sveta”, koje ne moraju biti, i najčešće nisu, u potpunosti međusobno jednake. Rešavanje problema duplikata nije jednostavan zadatak jer reprezantacije nisu identične i nije ih moguće povezati, tj. utvrditi koji objekat opisuju, pomoću nekog jedinstvenog identifikatora objekta (primarni ključ u bazama podataka, URI, JMBG osobe, ISBN knjige i sl.). Problem je uočen i prvi put razmatran pre više od pola veka [8]. Newcombe [17] 1959. godine postavlja teorijske osnove problema, a 1969. godine Fellegi i Sunter daju i formalni model [10]. Terminologija nije standardizovana, pa se ovaj problem „krije” pod raznim nazivima. U statistici, najčešće se, osim izvornog termina *record linkage*, koristi i *record matching*, dok je, generalno, problem poznati još i kao: *object/instance identification*, *entity/name/record/coreference resolution*, *data-/record/fuzzy/approximate matching*, *deduplication*, *duplicate (record) detection*, itd.

Otkrivanje duplikata predstavlja važan korak pri integraciji podataka koji potiču iz više distribuiranih i/ili heterogenih izvora. Međutim, primetimo da se duplikati mogu javiti i ukoliko podaci potiču iz samo jednog izvora, kao posledica greške i različitih zapisa. Na primer, u jednom XML dokumentu naziv kolekcije knjiga može biti naveden kao

```
<title>
  Knjige iz Prvog svetskog rata, Narodna biblioteka Srbije
</title>
```

a u drugom dokumentu kao

```
<title>
  Kolekcija knjiga iz Prvog svetskog rata iz Narodne biblioteke
</title>
```

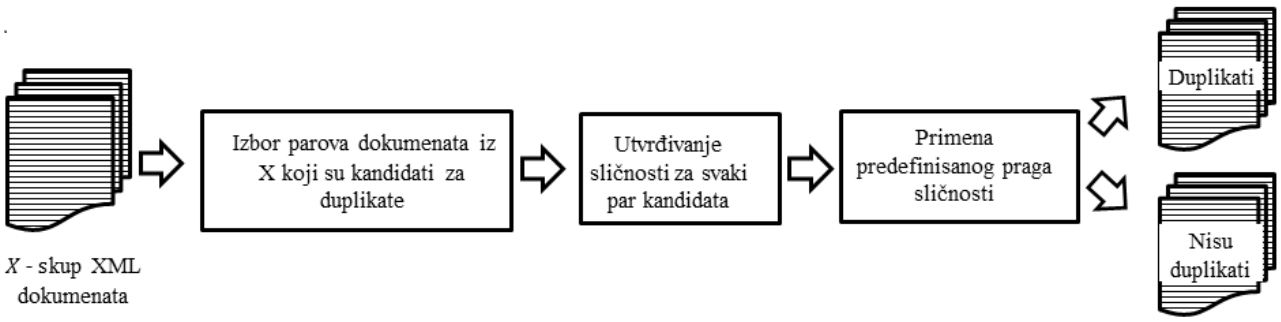
Otkrivanjem i uklanjanjem duplikata poboljšava se kvalitet podataka i otklanjaju greške koje su posledica višestrukih opisa (nekompletnost prikaza, netačne statistike, itd.). Na primer, u prisustvu duplikata, rezultati pretrage po kolekcijama bi, nekorektno, ukazivali na to da postoje dve različite kolekcija o Prvom svetskom ratu koje se nalaze u Narodnoj biblioteci Srbije.

Problem otkrivanja duplikata je dobro poznat i intenzivno proučavan poslednjih decenija u oblasti relacionih baza podataka. Detaljan pregled oblasti i metoda za otkriva-

nje duplikata u relacionim bazama dat je u [9]. Sa nastankom XML-a, krajem devedesetih godina XX veka, i njegovom velikom popularnošću koja je rezultovala ogromnim količinama podataka predstavljenim u ovom formatu, rešavanje problema višestrukih XML reprezentacija postaje popularna tema za istraživanje u poslednjih desetak godina. XML dokumente karakteriše hijerarhijska organizacija, kao što smo videli u Poglavljima 2 i 3, kompleksnost strukture, kao i mogućnost postojanja većih razlika u strukturi čak i između dokumenata koji su u skladu sa istim DTD ili XML shemom (usled elemenata koji su definisani kao opcioni, elemenata koji se mogu ponavljati, a čiji broj ponavljanja nije fiksiran, itd.), što iziskuje nešto drugačije metode za rešavanje ovog problema.

Većina algoritama za otkrivanje XML duplikata zasniva se na poređenju sličnosti parova dokumenata, pri čemu su poređenja međusobno nezavisna. Ovakvi algoritmi se u literaturi sreću pod nazivom **iterativni**. Sastoje se od tri glavna koraka koji su prikazani na Slici 4. Prvi korak se sastoji od izbora parova XML dokumenata koji su potencijalni duplikati. Kako broj XML dokumenata može biti jako veliki (stotine hiljada ili milioni dokumenata), u ovom koraku vrši se grubo grupisanje XML dokumenata i smanjuje broj parova za koje se precizno utvrđuje sličnost. Tim postupkom se značajno može povećati efikasnost celokupnog procesa. Ovo inicijalno filtriranje parova potrebno je realizovati na način koji je jednostavan i nije „skup”. Tako, recimo, iako opisi kolekcija knjiga mogu biti prikupljeni iz različitih izvora, svaka kolekcija se (fizički) nalazi u tačno jednoj instituciji. Odavde se zaključuje da ima smisla tražiti duplikate samo među onim XML dokumentima koji opisuju kolekcije iz iste institucije. Stoga se skup svih dokumenata može predstaviti pomoću particija P_n, pri čemu svaka particija odgovara jednoj instituciji, a kandidati za duplikate bi bili parovi dokumenata (x_i, x_j) takvi da $x_i, x_j \in P_n$. Nakon odabira kandidata, u drugom koraku se za svaki odabrani par utvrđuje sličnost primenom odabrane mere sličnosti. Zatim se, u trećem koraku, u slučaju da je sličnost između dokumenata veća od neke unapred zadate vrednosti, dokumenti klasifikuju kao duplikati, a u suprotnom se smatra da nisu duplikati. Primetimo da je relacija „biti duplikat”, relacija ekvivalencije i da se, nakon što je u trećem koraku utvrđeno koji dokumenti predstavljaju duplikate, može odrediti tranzitivno zatvorenje, čime bi polazni skup XML dokumenata bio podeljen na klase ekvivalencije od kojih bi svaka sadržala one dokumente koji opisuju jedan isti objekat.

Nakon identifikacije duplikata često sledi proces njihovog spajanja (eng. *data fusion*, *data merge*). U procesu spajanja veoma često je neophodno razrešiti konflikte i probleme nekonzistentnih podataka. Tako, u jednom opisu kolekcije kao datum objavljivanja može stajati `<date>2014</date>`, dok drugi opis može sadržati drugačiju informaciju `<date>2013-11-28</date>`. Problem razrešavanja konflikata i efektivno spajanje pronađenih duplikata izlazi van okvira ovog rada, u kome se bavimo samo otkrivanjem duplikata, što predstavlja prvi korak u integraciji podataka, i upućujemo na pregledni rad o *data fusion* metodama [3].



Slika 4: Glavni koraci u procesu otkrivanja duplikata.

5.1 Homogeni i heterogeni duplikati

Reprezentacije objekata, u našem slučaju XML dokumenti, prikupljeni iz više izvora mogu biti u skladu sa istim DTD, odnosno XML shemom, i tada govorimo o homogenim duplikatima, ili se pak mogu bitno razlikovati po strukturi, i tada govorimo o problemu heterogenih duplikata.

Primeri 7 i 8 prikazuju metapodatke o kolekciji koja se nalazi u Narodnoj biblioteci Srbije, a koju čine dokumenti iz perioda Prvog svetskog rata. Oba opisa kolekcije su u skladu sa istom XML shemom i prate isti standard za predstavljanje metapodataka, Dublin Core (DC)[2]. Primetimo, međutim, da se sadržaj opisa razlikuje, npr. u XML-u iz Primera 7 nema tekstualnog opisa kolekcije koji je dat u XML-u iz Primera 8 u elementu `<dc:description>`. Utvrđivanjem da ova dva dokumenta opisuju isti objekat iz „stvarnog sveta”, tj. istu kolekciju, i njihovim spajanjem u novi Dublin Core opis koji sadrži sve informacije prisutne u polaznim dokumentima (videti Primer 9), izbegla bi se greška dupliranja kolekcije koja može dovesti do nekorektnih rezultata pretrage (u kojima bi duplirani opisi bili predstavljeni kao različiti entiteti), grešaka u statistikama i izveštajima koji prikazuju broj postojećih kolekcija i u isto vreme dao potpuniji opis kolekcije.

Primer 7. Metapodaci u Dublin Core formatu o kolekciji dokumenata iz Prvog svetskog rata koja se nalazi u Narodnoj biblioteci Srbije (izvor 1).

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>Letters, postcards, unpublished memoirs and documents
    for the history of Serbia in World War I</dc:title>
  <dc:creator>Narodna biblioteka Srbije</dc:creator>
  <dc:subject>Policy</dc:subject>
  <dc:subject>Social History</dc:subject>
  <dc:subject>Community Studies</dc:subject>
  <dc:subject>World War, 1914-1918</dc:subject>
  <dc:format>unknown</dc:format>
  <dc:source>National library of Serbia
     http://www.nb.rs/?change_lang=en</dc:source>
```

```
    <dc:language>srp</dc:language>
    <dc:rights>Freely available online</dc:rights>
  </oai_dc:dc>
```

Primer 8. Metapodaci u Dublin Core formatu o kolekciji dokumenata iz Prvog svetskog rata koja se nalazi u Narodnoj biblioteci Srbije (izvor 2).

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>Letters, postcards, unpublished memoirs and documents
    for the history of Serbia in World War I</dc:title>
  <dc:creator>NBS</dc:creator>
  <dc:creator>National Library of Serbia</dc:creator>
  <dc:creator>Narodna biblioteka Srbije</dc:creator>
  <dc:subject>Policy</dc:subject>
  <dc:subject>Social History</dc:subject>
  <dc:subject>Community Studies</dc:subject>
  <dc:subject>World War, 1914-1918</dc:subject>
  <dc:description>
    This Digital Collection provides particularly valuable letters...
  </dc:description>
  <dc:format>unknown</dc:format>
  <dc:language >srp</dc:language>
  <dc:coverage>Serbia</dc:coverage>
  <dc:rights>Freely available online</dc:rights>
</oai_dc:dc>
```

Primer 9. Spajanjem dupliranih opisa kolekcije kao rezultat dobija se jedna, kompletnija reprezentacija polazne kolekcije.

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
 <dc:title>Letters, postcards, unpublished memoirs and documents
  for the history of Serbia in World War I</dc:title>
 <dc:creator>Narodna biblioteka Srbije</dc:creator>
 <dc:subject>Policy</dc:subject>
 <dc:subject>Social History</dc:subject>
 <dc:subject>Community Studies</dc:subject>
 <dc:subject>World War, 1914-1918</dc:subject>
 <dc:format>unknown</dc:format>
 <dc:source>National library of Serbia
     http://www.nb.rs/?change_lang=en</dc:source>
 <dc:language >srp</dc:language>
 <dc:rights>Freely available online</dc:rights>
</oai_dc:dc>

<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
 <dc:title>Letters, postcards, unpublished memoirs and documents
  for the history of Serbia in World War I</dc:title>
 <dc:creator>NBS</dc:creator>
 <dc:creator>National Library of Serbia</dc:creator>
 <dc:creator>Narodna biblioteka Srbije</dc:creator>
 <dc:subject>Policy</dc:subject>
 <dc:subject>Social History</dc:subject>
 <dc:subject>Community Studies</dc:subject>
 <dc:subject>World War, 1914-1918</dc:subject>
 <dc:description>
  This Digital Collection provides particularly valuable letters...
 </dc:description>
 <dc:format>unknown</dc:format>
 <dc:language >srp</dc:language>
 <dc:coverage>Serbia</dc:coverage>
 <dc:rights>Freely available online</dc:rights>
</oai_dc:dc>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
 <dc:title>Letters, postcards, unpublished memoirs and documents
  for the history of Serbia in World War I</dc:title>
 <dc:creator>NBS</dc:creator>
 <dc:creator>National Library of Serbia</dc:creator>
 <dc:creator>Narodna biblioteka Srbije</dc:creator>
 <dc:subject>Policy</dc:subject>
 <dc:subject>Social History</dc:subject>
 <dc:subject>Community Studies</dc:subject>
 <dc:subject>World War, 1914-1918</dc:subject>
 <dc:description>
  This Digital Collection provides particularly valuable letters...
 </dc:description>
 <dc:format>unknown</dc:format>
 <dc:source>National library of Serbia
     http://www.nb.rs/?change_lang=en</dc:source>
 <dc:language >srp</dc:language>
 <dc:coverage>Serbia</dc:coverage>
 <dc:rights>Freely available online</dc:rights>
</oai_dc:dc>
```

Sa druge strane, ista kolekcija je u Primeru 10 data u skladu sa Encoded Archival Description(EAD)[2] standardom za metapodatke. U ovom slučaju, Dublin Core opis iz Primera 7 (odnosno 8) i EAD opis su duplikati i to heterogeni. I u slučaju heterogenih duplikata moguće je konstruisati novi, jedinstven opis koji bi sadržao integrisane sve informacije iz polaznih duplikata, međutim, druga mogućnost, koja je za primere heterogenih duplikata koje smo naveli prirodnija, korisnija i znatno lakša za realizaciju, je da se prepoznati duplikati zadrže i povežu kao različiti pogledi na istu kolekciju. Ovakvim povezivanjem dobio bi se bogatiji i kompletniji opis polazne kolekcije.

Primer 10. Metapodaci u EAD formatu o kolekciji dokumenata iz Prvog svetskog rata koja se nalazi u Narodnoj biblioteci Srbije (izvor 3).

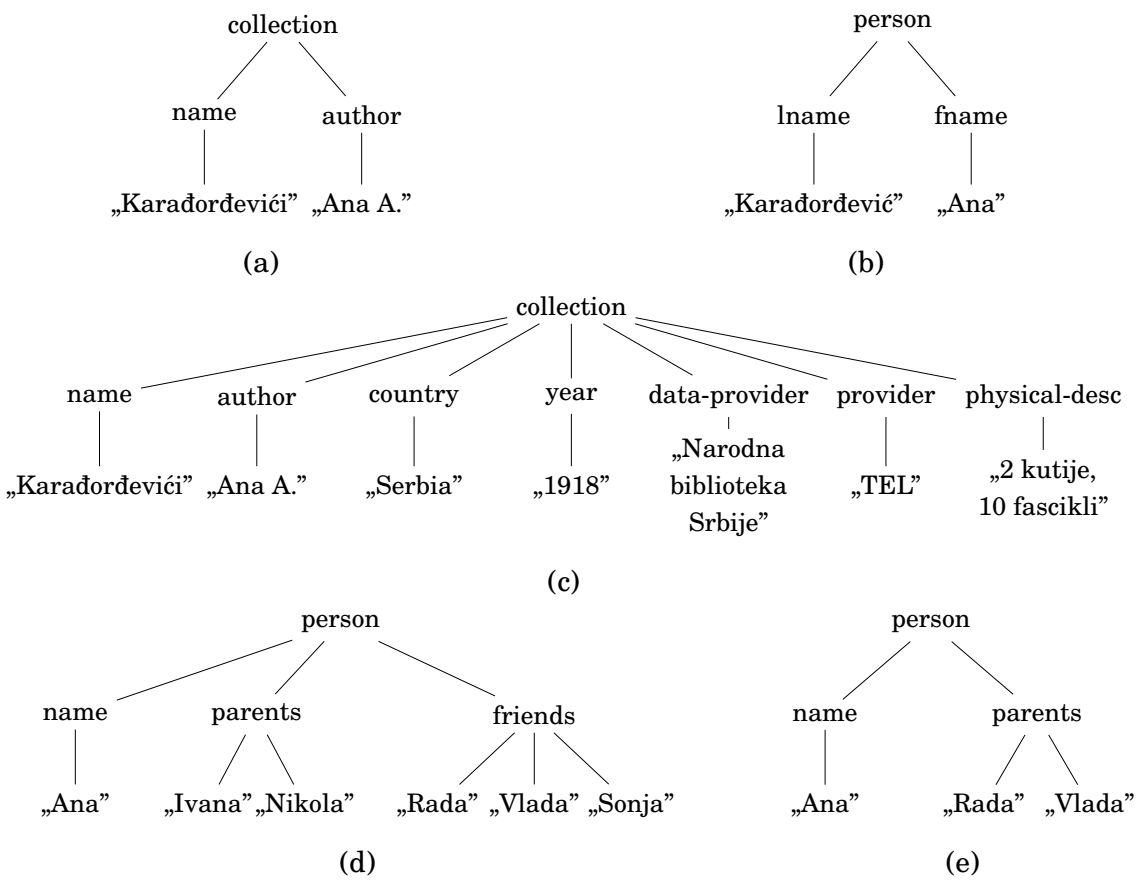
```
<?xml version="1.0" encoding="UTF-8"?>
<ead>
  <eadheader>
    <eadid
       identifier="documents-for-history-of-serbia-in-world-war-i">a0569</eadid>
    <filedesc>
      <titlestmt>
        <titleproper encodinganalog="title">Letters, postcards, unpublished
           memoirs and documents
          for the history of Serbia in World War I</titleproper>
      </titlestmt>
      <publicationstmt>
        <publisher encodinganalog="publisher">Narodna biblioteka
           Srbije</publisher>
        <address>
           <addressline>Skerliceva 1</addressline>
           <addressline>Beograd</addressline>
           <addressline>Serbia</addressline>
           <addressline>11000</addressline>
           <addressline>Telephone: +381 11 2451242</addressline>
           <addressline>Email: nbs@nb.rs</addressline>
           <addressline>www.nb.rs</addressline>
        </address>
        <date normal="2013-11-28" encodinganalog="date">2013-11-28</date>
      </publicationstmt>
    </filedesc>
  </eadheader>
  <archdesc level="collection" relatedencoding="ISAD(G)v2">
    <did>
      <unittitle encodinganalog="3.1.2">Letters, postcards, unpublished
         memoirs and documents
        for the history of Serbia in World War I</unittitle>
      <unitid encodinganalog="3.1.1" countrycode="RS">a0569</unitid>
      <unitdate normal="1914/1918" encodinganalog="3.1.3">1914 -
         1918</unitdate>
      <physdesc encodinganalog="3.1.5">
```

```
        Digital images of letters and correspondence, scripts, diaries,
            postcard</physdesc>
      <langmaterial encodinganalog="3.4.3">
        <language langcode="srp">Serbian</language>
      </langmaterial>
      <note type="generalNote">
        <p>The materials have been digitized within the context of Europeana
            Collections 1914-1918 </p>
      </note>
      <note type="generalNote">
        <p>http://www.europeana-collections-1914-1918.eu/</p>
      </note>
    </did>
  </archdesc>
</ead>
```

5.2 Primer algoritma za otkrivanje XML duplikata

Posmatrajmo primere stabala date na Slici 5. Primetimo da stabla (a) i (c) opisuju isti objekat, jednu kolekciju materijala o dinastiji Karađorđević, s tim što stablo (c) sadrži dodatne informacije koje nisu date stablom (a) (zemlju i godinu nastanka, itd.). Sa druge strane, stablo (b) opisuje drugi objekat, jednu osobu. Međutim, primenom *tree edit distance* mere dobija se da je rastojanje između stabala (a) i (b) manje od rastojanja između (a) i (c), tj. da je XML dokument predstavljen stablom (a) sličniji dokumentu datim stablom (b). Razmotrimo dalje stabla (d) i (e). Najmanje rastojanje između stabala dobija se tako što se u stablu (d) obriše podstablo sa korenom u `parents`, čvor `friends` preimenuje u `parents`, i, na kraju, obriše tekstualni čvor „`Sonja`". Ovim se, očigledno narušava semantika sadržana u nazivima elemenata. Osim toga, kao što smo napomenuli u Poglavlju 3, redosled između dece istog čvora često nije značajan i uređenje na skupu čvorova je zapravo parcijalno. Dok za uređena stabla *tree edit distance* problem ima polinomijalnu složenost, za neuređena stabla problem postaje NP-težak [22]. Navedeni nedostaci motivisali su autore u [15] da predlože meru koja bi omogućila prevazilaženje uočenih problema koji se javljaju kada se na XML stabla primeni klasična *tree edit distance* mera.

Nova mera zasniva se ideji **preklapanja** (eng. *overlay*) XML stabala. Ukoliko dva stabla imaju potpuno istu strukturu i razlikuju se samo u tekstualnim vrednostima sadržanim u listovima, tada se ona mogu preklopiti uparivanjem čvorova koji imaju isti put od korenskog čvora. Ukoliko je moguće realizovati ovakvo preklapanje na više načina, tada se bira ono za koje je razlika u tekstualnim vrednostima uparenih listova najmanja. Za ocenu razlike između tekstualnih vrednosti koristi se neka od poznatih mera rastojanja između niski [9].



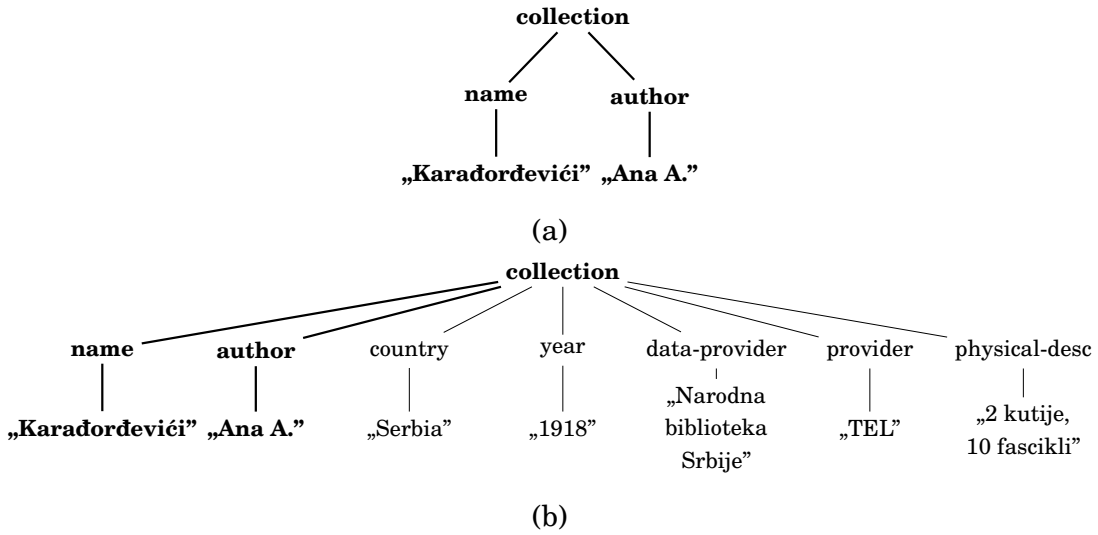
Slika 5: Problemi koji se javljaju pri poređenju XML dokumenata pomoću klasične *tree edit distance* mere.

Kroz primere ćemo koristiti Levenštajnovo rastojanje [14] kao jednu od osnovnih i najjednostavnijih mera. Levenštajnovo rastojanje između niski a i b označavamo sa $lev_{a,b}(|a|,|b|)$, gde su $|a|$ i $|b|$ dužine niski, i računamo na sledeći način:

$$lev_{a,b}(i,j) = \begin{cases} i & \text{, ako je } j = 0 \\ j & \text{, ako je } i = 0 \\ min \begin{cases} lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{, inače} \end{cases}$$

gde su a_i, odnosno b_j, i-ti, odnosno j-ti karakter odgovarajućih niski, $1_{(a_i \neq b_j)} = 0$ ako je $a_i = b_j$, a u suprotnom je $1_{(a_i \neq b_j)} = 1$, i $lev_{a,b}(i,j)$ predstavlja rastojanje između prvih i karaktera u a i prvih j karaktera u b.

Ukoliko postoje razlike u strukturi stabala, usled dodatnih informacija u jednom stablu, koje ne postoje u drugom, preklapanje se i dalje može realizovati, brisanjem podstabala koja sadrže te dodatne informacije i koja se ne mogu upariti. Na Slici 6 prikazano je jedno preklapanje stabala koje smo prethodno razmatrali.



Slika 6: Preklapanje (eng. *overlay*) XML stabala.

Definišimo sada i formalno pojam **preklapanja**.

Definicija 15. Neka su $T_1 = (N_1, E_1, <_1, \lambda_1, v_1)$ i $T_2 = (N_2, E_2, <_2, \lambda_2, v_2)$ dva XML stabla. Preklapanje O stabala T_1 i T_2 je skup $O \subseteq N_1 \times N_2$, takav da:

(i) $O \neq \emptyset$

(ii) za sve $(n, m), (n', m') \in O$ važi $n = n'$ akko $m = m'$;

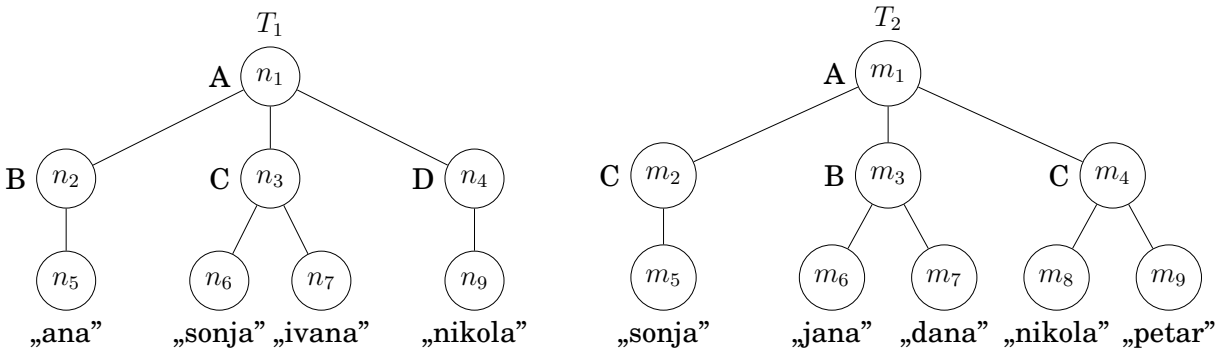
(iii) za sve $(n, m) \in O$ važi $path_{T_1}(n) = path_{T_2}(m)$,
gde $path_{T_i}(x)$ označava niz labela $\lambda_i(r_i), ..., \lambda_i(x)$ na putu od korena r_i do čvora $x \in N_i$, za $i \in \{1, 2\}$;

(iv) za sve $n \in N_1 \setminus Txt_{T_1}, m \in N_2 \setminus Txt_{T_2}$, važi $(n, m) \in O$ ako i samo ako postoje $n' \in N_1, m' \in N_2$
takvi da $(n, n') \in E_1$, $(m, m') \in E_2$ i $(n', m') \in O$.

Kažemo da su čvorovi n i m **upareni** ili **pridruženi** kada $(n, m) \in O$. Ukoliko nije uparen ni sa jednim drugim čvorom, čvor se briše. Definicija 15 zapravo kaže da: (ii) uparivanje čvorova je jedinstveno, (iii) čvorovi se mogu upariti jedino ako im se putevi od korena poklapaju, i (iv) ako su čvorovi, koji nisu listovi, upareni, tada oni imaju naslednike koji su takođe upareni. Primetimo da na osnovu (iv) sledi da ukoliko je čvor obrisan, tada su i svi njegovi naslednici obrisani. Ovo takođe znači i da se za dva stabla T_1 i T_2 može naći preklapanje jedino ako postoje dva lista $n \in Txt_{T_1}$ i $m \in Txt_{T_2}$ koji imaju isti put od korena.

Definicija 16. Za dva stabla kažemo da su **uporediva** kada postoji bar jedno njihovo preklapanje.

Definicija 17. Preklapanje O stabala T_1 i T_2 je **maksimalno** ukoliko ne postoji nijedno drugo preklapanje O' ovih stabala takvo da je $O \subset O'$.



Slika 7: Jedno maksimalno preklapanje stabala T_1 i T_2 dato je sa $O' = \{(n_1, m_1), (n_2, m_3), (n_3, m_2), (n_5, m_6), (n_6, m_5)\}$.

Na Slici 7 data su dva stabla T_1 i T_2. Jedno preklapanje ovih stabala je

$$O = \{(n_1, m_1), (n_3, m_2), (n_6, m_5)\}.$$

Ovo preklapanje nije maksimalno jer možemo naći druga preklapanja koja ga sadrže. Jedno takvo je

$$O' = \{(n_1, m_1), (n_2, m_3), (n_3, m_2), (n_5, m_6), (n_6, m_5)\}.$$

Intuitivno, O' je maksimalno jer ne možemo više upariti nove čvorove, a da ne narušimo svojstva preklapanja data Definicijom 15. Primetimo, međutim, da maksimalno preklapanje ne mora biti jedinstveno. Za stabla na Slici 7 još neka maksimalna preklapanja data su sa

$$O'' = \{(n_1, m_1), (n_2, m_3), (n_3, m_2), (n_5, m_7), (n_6, m_5)\}$$

kao i sa

$$O''' = \{(n_1, m_1), (n_2, m_3), (n_3, m_4), (n_5, m_6), (n_6, m_8), (n_7, m_9)\}.$$

Definicija 18. Neka su T_1 i T_2 XML stabla i neka je sa $sdest$ označena mera rastojanja niski. Tada se **cena uparivanja** μ^{sdest} dva čvora određuje na sledeći način:

$$\mu^{sdest}(n, m) = \begin{cases} sdest(v_1(n), v_2(m)) & \text{, ako su } n, m \text{ listovi} \\ 0 & \text{, inače} \end{cases}$$

Definicija 19. Neka su T_1 i T_2 XML stabla i neka je O jedno njihovo preklapanje. Neka dalje $sdest$ označava rastojanje između niski, a μ^{sdest} odgovarajuću cenu uparivanja čvorova. Tada se **cena preklapanja** $\Gamma^{sdest}(O)$ određuje na sledeći način:

$$\Gamma^{sdest}(O) = \sum_{(n,m) \in O} \mu^{sdest}(n, m).$$

Definicija 20. Preklapanje O stabala T_1 i T_2 je **optimalno**, za izabranu meru $sdest$, ako je maksimalno, i ukoliko ne postoji drugo maksimalno preklapanje O' stabala T_1 i T_2 takvo da je $\Gamma^{sdest}(O') < \Gamma^{sdest}(O)$.

Ukoliko je iz konteksta jasno koja mera rastojanja između niski se koristi, ili u slučaju da odabir ove mere nije od značaja, pisaćemo samo μ, odnosno Γ.

Primenom Levenštajnovog rastojanja lev, za stabla na Slici 7 izračunavamo cenu maksimalnog preklapanja $\Gamma^{lev}(O')$, gde je $O' = \{(n_1, m_1), (n_2, m_3), (n_3, m_2), (n_5, m_6), (n_6, m_5)\}$ kao:

$$\begin{aligned}\Gamma^{lev}(O') &= \mu^{lev}(n_1, m_1) + \mu^{lev}(n_2, m_3) + \mu^{lev}(n_3, m_2) + \mu^{lev}(n_5, m_6) + \mu^{lev}(n_6, m_5) \\ &= 0 + 0 + 0 + lev(\text{„ana”}, \text{„jana”}) + lev(\text{„sonja”}, \text{„sonja”}) \\ &= 1.\end{aligned} \tag{1}$$

Na isti način dobijamo da je cena preklapanja $\Gamma^{lev}(O'')$ gde je $O'' = \{(n_1, m_1), (n_2, m_3), (n_3, m_2), (n_5, m_7), (n_6, m_5)\}$, takođe 1. Izračunavanjem cene svih maksimalnih preklapanja stabala T_1 i T_2 može se utvrditi da su upravo O' i O'' maksimalna preklapanja sa najmanjom cenom, odakle zaključujemo da su O' i O'' upravo i optimalna preklapanja. Jednostavno se pokazuje da, ukoliko su dva stabla T_1 i T_2 uporediva, tada postoji i bar jedno maksimalno, odnosno optimalno preklapanje za T_1 i T_2.

Narednom definicijom autori u [15] uvode novu meru koja koristi pojam preklapanja između XML stabala kojim bi se izbegli pretodno uočeni problemi.

Definicija 21. Neka su T_1 i T_2 uporediva XML stabla. Tada je rastojanje između T_1 i T_2 jednako ceni njihovog optimalnog preklapanja.

Primetimo da se primenom pojma preklapanja i upravo uvedene mere na stabla sa Slike 5 utvrđuje da stabla (a) i (b) nisu uporediva, da je rastojanje između stabala (a) i (c) nula, a da pri poređenju stabala (d) i (e) ne dolazi do uparivanja onih unutrašnjih čvorova koji sadrže (semantički) različite informacije.

Pre nego što prikažemo algoritam za efektivno utvrđivanje rastojanja između XML stabala koji je predložen u [15], pokazaćemo neka svojstva preklapanja iz kojih sledi korektnost predloženog rešenja. Dokazi su izvorno u [15] dati u skraćenom obliku, u vidu ideja i smernica. Ovde ih izvodimo detaljno, osim u onim slučajevima kada trivijalno slede na osnovu prethodno pokazanog i samih definicija.

Teorema 1. *Neka je O jedno preklapanje stabala T_1 i T_2. Ako $(n, m) \in O$, tada je skup $O_{n,m} = \{(n', m') \in O | n' \in N_{T_1(n)}$ i $m' \in N_{T_2(m)}\}$ jedno preklapanje stabala $T_1(n)$ i $T_2(m)$, gde je sa $T_1(n)$ označeno podstablo stabla T_1 sa korenom u n, a sa $N_{T_1(n)}$ skup čvorova stabla $T_1(n)$ (i analogno za stablo T_2 i čvor m). Štaviše, ako je O maksimalno preklapanje, tada je i $O_{n,m}$ takođe maksimalno. Za preklapanje $O_{n,m}$ kažemo da je indukovano na $T_1(n)$ i $T_2(m)$ preklapanjem O.*

Dokaz. Pokažimo da za skup $O_{n,m} \subset O$ važe svojstva (i)-(iv) iz Definicije 15.

(i) Kako $(n, m) \in O_{n,m}$, zaključujemo da važi $O_{n,m} \neq \emptyset$.

(ii) Pokažimo da za svako $(n', m'), (n'', m'') \in O_{n,m}$ važi uslov:

$$n' = n'' \text{ akko } m' = m'' \tag{2}$$

Dokaz izvodimo svođenjem na kontradikciju. Pretpostavimo da postoje takvi $(n', m'), (n'', m'') \in O_{n,m}$ za koje ne važi 2. Kako je $O_{n,m} \subset O$, imamo $(n', m'), (n'', m'') \in O$, odakle sledi da ni O nije preklapanje $\lightning$.

(iii) Pokažimo da za svako $(n', m') \in O_{n,m}$ važi:

$$path_{T_1(n)}(n') = path_{T_2(m)}(m') \tag{3}$$

Dokaz izvodimo svođenjem na kontradikciju. Pretpostavimo da postoje takvi $(n', m') \in O_{n,m}$ za koje ne važi uslov 3. Kako je $O_{n,m} \subset O$, imamo $(n', m') \in O$. Kako su $path_{T_1(n)}(n')$ i $path_{T_2(m)}(m')$ sufiksni podnizovi nizova $path_{T_1}(n')$ i $path_{T_2}(m')$ jednakih dužina, iz $path_{T_1(n)}(n') \neq path_{T_2(m)}(m')$ sledi $path_{T_1}(n') \neq path_{T_2}(m')$. Odavde zaključujemo da ni O nije preklapanje $\lightning$.

(iv) Pokažimo na kraju da za svaki čvor $n' \in N_{T_1(n)}$ i svaki $m' \in N_{T_2(m)}$, koji nisu listovi, važi:

$$\begin{aligned} &(n', m') \in O_{n,m} \text{ akko postoje } n'' \in N_{T_1(n)} \text{ i } m'' \in N_{T_2(m)} \text{ takvi da je} \\ &n' = parent(n'') \text{ i } m' = parent(m'') \text{ i } (n'', m'') \in O_{n,m} \end{aligned} \tag{4}$$

Pretpostavimo prvo da $(n', m') \in O_{n,m}$ i pokažimo da postoje čvorovi n'' i m'' takvi da važi 4. Kako je $(n', m') \in O$ zaključujemo da postoje $(n'', m'') \in O$ takvi da važi 4. Kako su n'' i m'' deca čvorova n' i m', iz $(n', m') \in O_{n,m}$ sledi $(n'', m'') \in O_{n,m}$, jer je $O_{n,m}$ indukovano preklapanjem O.

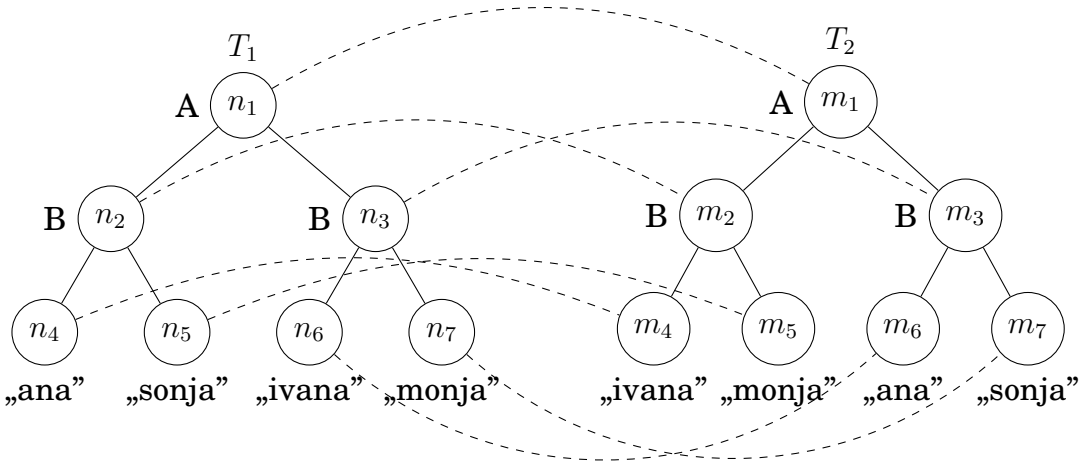
Pokažimo da važi i obrnuto. Iz $(n'', m'') \in O_{n,m}$ imamo $(n'', m'') \in O$, a kako je po pretpostavci $n' = parent(n'')$ i $m' = parent(m'')$, sledi $(n', m') \in O$ jer je O preklapanje. Pošto je $n' \in N_{T_1(n)}$ i $m' \in N_{T_2(m)}$, zaključujemo da $(n', m') \in O_{n,m}$.

Pokažimo na kraju da, ukoliko je O maksimalno, i $O_{n,m}$ je takođe maksimalno. Dokaz izvodimo svođenjem na kontradikciju. Pretpostavimo da preklapanje $O_{n,m}$ nije maksimalno. Tada postoji preklapanje $O'_{n,m}$ za $T_1(n)$ i $T_2(m)$ takvo da je $O_{n,m} \subset O'_{n,m}$, iz čega sledi da postoji preklapanje O' za T_1 i T_2 takvo da važi $O \subset O'$. Odavde zaključujemo da preklapanje O nije maksimalno $\lightning$.

□

Teorema 2. *Neka je O jedno preklapanje stabala T_1 i T_2. Tada je:*

$$O = (r_1, r_2) \cup \bigcup_{\substack{n \in child(r_1) \\ m \in child(r_2) \\ (n,m) \in O}} O_{n,m} \tag{5}$$



Slika 8: Preklapanje O stabala T_1 i T_2 koje nije optimalno, iako su njim indukovana preklapanja O_{n_2,m_2} i O_{n_3,m_3} optimalna.

Dokaz. Na osnovu definicije indukovanog preklapanja važi $O_{n,m} \subset O$, za svako $n \in child(r_1)$, $m \in child(r_2)$. Kako je i $(r_1, r_2) \in O$, zaključujemo

$$O \supseteq (r_1, r_2) \cup \bigcup_{\substack{n \in child(r_1) \\ m \in child(r_2) \\ (n,m) \in O}} O_{n,m}.$$

Za svako $(n', m') \in O$ jednostavno se pokazuje da važi $(n', m') \in (r_1, r_2) \cup \bigcup_{\substack{n \in child(r_1) \\ m \in child(r_2) \\ (n,m) \in O}} O_{n,m}$. □

Teorema 3. *Neka je O jedno preklapanje stabala T_1 i T_2. Tada je:*

$$\Gamma(O) = \sum_{\substack{n \in child(r_1) \\ m \in child(r_2) \\ (n,m) \in O}} \Gamma(O_{n,m})$$

Dokaz. Dokaz sledi direktno iz Teoreme 2 i definicije cene preklapanja. □

Iz prethodno pokazanog i na osnovu definicije cene preklapanja, trivijalno se pokazuje da, ako je preklapanje O optimalno, tada su i sva njime indukovana preklapanja $O_{n,m}$ takođa optimalna, gde je n, odnosno m, dete korenskog čvora r_1, odnosno r_2. Međutim, obrnuto ne važi, što ćemo pokazati narednim primerom. Posmatrajmo preklapanje O dato na Slici 8. Iako su njim indukovana preklapanja podstabala $T_1(n_2)$ i $T_2(m_2)$, odnosno $T(n_3)$ i $T(m_3)$, optimalna, jednostavno se pokazuje da O nije optimalno:

$$\begin{aligned}\Gamma^{lev}(O) &= lev(\text{„ana”}, \text{„ivana”}) + lev(\text{„sonja”}, \text{„monja”}) \\ &\quad + lev(\text{„ivana”}, \text{„ana”}) + lev(\text{„monja”}, \text{„sonja”}) \\ &= 2 + 1 + 2 + 1 \\ &= 6.\end{aligned}$$

Uočimo sada preklapanje $O' = \{(n_1, m_1), (n_2, m_3), (n_3, m_2), (n_4, m_6), (n_5, m_7), (n_6, m_4), (n_7, m_5)\}$. Cena preklapanja O' je:

$$\begin{aligned}\Gamma^{lev}(O') &= lev(\text{„ana”}, \text{„ana”}) + lev(\text{„sonja”}, \text{„sonja”}) \\ &+ lev(\text{„ivana”}, \text{„ivana”}) + lev(\text{„monja”}, \text{„monja”}) \\ &= 0.\end{aligned}$$

Očigledno je da je O' jedno maksimalno preklapanje stabala T_1 i T_2, a na osnovu upravo izračunate cene, zaključujemo i da je optimalno. Pre nego što odredimo potreban i dovoljan uslov za optimalnost preklapanja, uvešćemo pojam **pridruživanje čvorova prvog nivoa**.

Definicija 22. Za data stabla T_1 i T_2, **pridruživanje čvorova prvog nivoa**, u oznaci A, je skup koji čine parovi čvorova (n, m), sa sledećim svojstvima:

- n je dete korenskog čvora stabla T_1,
- m je dete korenskog čvora stabla T_2,
- Stabla $T_1(n)$ i $T_2(m)$ su uporediva,
- Pridruživanje je jednoznačno, tj. za svako $(n, m), (n', m') \in A$ važi $n = n'$ akko $m = m'$.

Pojmovi maksimalnosti, cene i optimalnosti, prethodno definisani za preklapanja, jednostavno se, po analogiji, definišu za ovako uveden pojam pridruživanja.

Definicija 23. Pridruživanje prvog nivoa A stabala T_1 i T_2 je **maksimalno** ukoliko ne postoji nijedno drugo pridruživanje A' ovih stabala takvo da je $A \subset A'$.

Jednostavno se pokazuje da važi:

Teorema 4. *Preklapanje je maksimalno ako i samo ako sadrži maksimalno pridruživanje prvog nivoa.*

Definicija 24. Cena pridruživanja prvog nivoa $\Gamma(A)$ određuje se na sledeći način:

$$\Gamma(A) = \sum_{(n,m) \in A} \Gamma(O_{n,m})$$

Definicija 25. Pridruživanje prvog nivoa A stabala T_1 i T_2 je optimalno ako je maksimalno i ukoliko ne postoji drugo maksimalno pridruživanje A' ovih stabala takvo da je $\Gamma(A') < \Gamma(A)$.

Iz Teoreme 4 i na osnovu definicija cene preklapanja i cene pridruživanja, sledi:

Teorema 5. *Preklapanje O je optimalno akko sadrži optimalno pridruživanje prvog nivoa.*

Iz prethodnih rezultata izvodi se efektivana procedura pronalaženje optimalnog preklapanja za dva zadata stabla. Nakon pridruživanja korenskih čvorora, određuju se sva maksimalna pridruživanja prvog nivoa i za svako od njih računa cena kako bi se odredilo optimalno pridruživanje. Izračunavanje cene i određivanje optimalnog pridruživanja izvodi se rekurzivno sve do listova. Moraju se odrediti sva maksimalna pridruživanja i izračunati njihove cene, kako bi se utvrdilo optimalno preklapanje. Pseudo-kod je dat Algoritmom 1.

Algoritam 1 Otkrivanje duplikata

```
 1: Input:
        (T, T') - par XML stabala koji su kandidati za duplikate,
        t - prag
 2: function DIST(T,T')
 3:     r ← root(T)
 4:     r' ← root(T')
 5:     if isLeaf(r) and isLeaf(r') then
 6:         return sdist(r,r')
 7:     else
 8:         xmldist ← ∞
 9:         for l in labels(children(r) ∪ children(r')) do
10:             for n_i in children_l(r) do
11:                 for m_j in children_l(r') do
12:                     D_l[i, j] ← DIST(T(n_i),T(m_j))
13:                 end for
14:             end for
15:             A_l ← findAssignment(D_l)
16:             for (i, j) in A_l do
17:                 if xmldist = ∞ then
18:                     xmldist ← 0
19:                 end if
20:                 xmldist ← xmldist + D_l[i, j]
21:             end for
22:             return xmldist
23:         end for
24:     end if
25: end function
26: if DIST(T,T') < t then
27:     return true
28: else
29:     return false
30: end if
```

Algoritam rekurzivno analizira stabla. Ukoliko su u pitanju listovi, porede se tekstualni sadržaji ovih čvorova pomoću neke od poznatih mera za izračunavanje rastojanja između niski [9]. U suprotnom, razmatraju se deca ovih čvorova i za svaki par čvorova koji imaju istu labelu rekurzivno se izračunava cena preklapanja podstabala čiji su oni koreni. Nakon što se odrede sve cene, vrši se uparivanje onih čvorova koji imaju iste labele i to na takav način da se minimizuje ukupna cena. Utvrđivanje traženog uparivanja je zapravo jedan problem dodeljivanja i može se rešiti npr. pomoću Mađarskog algoritma [16]. U Algoritmu 1, ovaj zadatak je označen pozivom procedure *findAssignment*. Za datu matricu koja sadrži rastojanja između čvorova, procedura vraća listu parova čvorova, tj. njihovih indeksa, čijim se uparivanjem dobija optimalno preklapanje. Radi jednostavnosti i čitljivosti, u pseudo-kodu smo skup koji sadrži decu čvora r koja imaju labelu l označavali pomoću $children_l(r)$. Rezultati izračunavanja rastojanja za jedan takav skup dece čija je labela l, čuvaju se u matrici D_l. Rastojanje se inicijalno postavlja na ∞, a resetuje se na nulu jedino ukoliko se nađe dodeljivanje između dece korenskih čvorova.

6 Zaključak

U ovom radu bavimo se problemom utvrđivanja sličnosti između XML dokumenata u cilju otkrivanja XML duplikata, tj. onih dokumenata koji opisuju isti objekat iz stvarnog sveta. Otkrivanje duplikata je važan korak u procesu integracije podataka i neophodan je kako bi se mogao pružiti potpuniji prikaz opisanih objekata i izbegle greške koje pri obradi podataka mogu nastati kao posledica duplikata. Kako XML dokumenti imaju hijerarhijsku strukturu i najčešće se predstavljaju pomoću stabala, prirodno je za određivanje sličnosti dva XML dokumenta koristiti meru rastojanja između stabala (*tree edit distance*). Ova mera, međutim, pretpostavlja uređena stabla, što XML stabla u opštem slučaju nisu. Osim toga, operacijom preimenovanja unutrašnjih čvorova mogu se narušiti semantičke informacije sadržane u nazivima XML elemenata i atributa kojima ovi čvorovi odgovaraju. Ovo je motivisalo autore u [15] da predlože novu meru zasnovanu na pojmu **preklapanja** (eng. *overlay*) kojom bi se otklonili uočeni nedostaci. U radu dajemo prikaz predložene mere i algoritma za otkrivanje XML duplikata koji je na njoj zasnovan.

Primetimo da, kako operacija preimenovanja unutrašnjih čvorova nije dopuštena, ovaj algoritam može otkriti samo homogene duplikate. Da bi se algoritam učinio primenjivim i za otkrivanje heterogenih duplikata, može se razmatrati upotreba ontologija i rečnika pomoću kojih bi bilo moguće utvrditi koji to elementi, iz različitih shema i sa različitim nazivima, nose semantički iste informacije.

Upoređivanje mera za određivanje sličnosti XML podataka i odgovarujićih algoritama za otkrivanje duplikata nije jednostavan zadatak. Uobičajeno je da su rešenja razvijana u specifičnom kontekstu i prilagođena specifičnim namenama i mogu dati neadekvatne rezultate kada se primene u drugačijim okvirima. Osim toga, ispitivanje i upoređivanje predloženih metoda često je teško ostvarivo usled nedostupnosti konkretnih detalja implementacije i kompletnih uzoraka podataka koji su korišćeni za testiranje i verifikaciju rešenja. Jedan pokušaj uvođenja standardizacije i prevazilaženja ovog problema dat je u [20].

7 Literatura

[1] Dublin Core Metadata Initiative. http://dublincore.org/.

[2] Encoded Archival Description. http://www.loc.gov/ead/.

[3] Jens Bleiholder and Felix Naumann. Data fusion. *ACM Computing Surveys (CSUR)*, 41(1):1, 2008.

[4] James Clark, Steve DeRose, et al. XML path language (XPath) version 1.0. http://www.w3.org/TR/xpath/, 1999.

[5] World Wide Web Consortium. Extensible markup language (XML). http://www.w3.org/TR/xml/, 1998.

[6] World Wide Web Consortium. Document Object Model (DOM) Level 3 Core Specification. http://www.w3.org/TR/DOM-Level-3-Core/, 2004.

[7] Theodore Dalamagas, Tao Cheng, Klaas-Jan Winkel, and Timos Sellis. A methodology for clustering XML documents by structure. *Information Systems*, 31(3):187–228, 2006.

[8] Halbert L Dunn. Record linkage*. *American Journal of Public Health and the Nations Health*, 36(12):1412–1416, 1946.

[9] Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16, 2007.

[10] Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[11] Charles F Goldfarb and Yuri Rubinsky. *The SGML handbook*. Oxford University Press, 1990.

[12] Giovanna Guerrini, Marco Mesiti, and Ismael Sanz. An overview of similarity measures for clustering XML documents. *Information Systems*, 2006.

[13] Nils Klarlund, Thomas Schwentick, and Dan Suciu. Xml: Model, schemas, types, logics, and queries. In Jan Chomicki, Ron van der Meyden, and Gunter Saake, editors, *Logics for Emerging Applications of Databases*, pages 1–41. Springer Berlin Heidelberg, 2004.

[14] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[15] Diego Milano, Monica Scannapieco, and Tiziana Catarci. Structure aware XML object identification. *IEEE Data Eng. Bull.*, 29(2):67–74, 2006.

[16] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[17] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage of vital records. *Science*, 130(3381):954–959, October 1959.

[18] Andrew Nierman and HV Jagadish. Evaluating structural similarity in XML documents. In *WebDB*, volume 2, pages 61–66, 2002.

[19] Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):422–433, 1979.

[20] Melanie Weis, Felix Naumann, and Franziska Brosy. A duplicate detection benchmark for XML (and relational) data. In *Proc. of Workshop on Information Quality for Information Systems (IQIS)*, 2006.

[21] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.

[22] Kaizhong Zhang, Rick Statman, and Dennis Shasha. On the editing distance between unordered labeled trees. *Information processing letters*, 42(3):133–139, 1992.

[23] Zhongping Zhang, Rong Li, Shunliang Cao, and Yangyong Zhu. Similarity metric for XML documents. In *Knowledge Management and Experience Management Workshop*, 2003.

Biografija

Milica Knežević rođena je u Bajinoj Bašti 20.3.1980. Gimnaziju „Josif Pančić” završila je 1999. godine. Diplomirala je 2005. na Matematičkom fakultetu u Beogradu, na smeru računarstvo, sa prosečnom ocenom 9,1. U oktobru 2015. upisuje master studije primenjene matematike na Fakultetu tehničkih nauka u Novom Sadu. Radila je kao programer na Matematičkom fakultetu i Beogradskoj berzi. Od 2011. godine zaposlena je na Matematičkom institutu SANU kao istraživač saradnik.