Representing and Reasoning on XML Documents: A Description Logic Approach

D. Calvanese, G. D. Giacomo, M. Lenzerini

Presented by Daisy Yutao Guo
University of Waterloo
March 4, 2003

Outline

- Introduction
- XML documents, DTDs and basic reasoning tasks
- The Description Logic (DL) for representing DTDs
- Representing and reasoning over DTDs in DL
- Retrieving XML documents from a document base
- Conclusion and discussion
Introduction

- Recent proposals suggest viewing the Web as a huge semistructured database
- Drawbacks: they pay insufficient attention to the representation and reasoning on document structures
- Importance of such reasoning facility: help in several tasks related to query processing; improve both precision and efficiency

Introduction (cont’d)

- Documents in the Web are described by means of ad-hoc languages, whose structure is typically explicitly marked up with special tags.
- XML is one of the most prominent formalism for defining marked-up documents.
- The structure of XML documents is described by means of DTDs.
- Types of reasoning about DTDs:
  - Conformance
  - Equivalence (Inclusion, Disjointness)
Introduction (cont’d)

- Contributions of this paper:
  - Present a formalization of XML DTDs in terms of an expressive Description Logic, called DL
  - Present sound, complete and terminating inference procedures for DL, which provide a reasoning mechanism to enable reasoning tasks on DTDs
    - Conformance: polynomial time
    - $R$-equivalence (inclusion, disjointness): exponential time
  - Propose a method for retrieving XML documents from a document base by taking advantage of DL

---

XML DTD and Document Instance Examples

```xml
<DOCTYPE Mail [ 
  <!ELEMENT Mail (From, To, (Subject)?, Body)> 
  <!ELEMENT From (Address)> 
  <!ELEMENT To (Address)> 
  <!ELEMENT Address (#PCDATA)> 
  <!ELEMENT Subject (#PCDATA)> 
  <!ELEMENT Body (#PCDATA | any)> 
]>

DTD $M$ for mail documents
```

```xml
<Mail>
  <From>
    <Address> Dente@dsn.fi.it</Address>
  </From>
  <To>
    <Address> Beatrice@pitti.fi.it</Address>
    <Address> Virgilio@lappr.rn.it</Address>
  </To>
  <Subject> Appointment </Subject>
  <Body>
    Why don’t we meet at disco inferno at midnight. Tel al Caronte. Cheers,
    - D.A.
  </Body>
</Mail>

A document instance conforming to the DTD
XML Documents

- Each XML document can be considered as a pair: (D, d):
  - D—DTD
  - d—document instance
- Alphabets:
  - \( T \)—terminals (e.g., #PCDATA)
  - \( E \)—element types. For each \( E \in E \), we associate a start tag \(<E> \) and end tag \(</E> \).

**Definition 1** The set \( \text{doc}_{T,E} \) of all possible document instances that can be built over \( T \) and \( E \) is defined inductively as follows:
  - If \( d \) is a terminal in \( T \), then \( d \in \text{doc}_{T,E} \).
  - If \( d \) is a sequence of the form \(<E>d_1\ldots d_k</E> \), where \( E \in E \) is an element type and \( d_1,\ldots,d_k \in \text{doc}_{T,E} \), then \( d \in \text{doc}_{T,E} \).

XML DTDs

- Symbol \( S \in T U E \), i.e., either a generic terminal or an element type
- **Definition 2**: A DTD \( D \) is a pair \((P, R)\):
  - \( R \): root element type, which specifies the document type
  - \( P \): a set of element type definitions, each in the form \( E -> \alpha \)
  - \( E \): the element type to be defined
  - \( \alpha \): the content model, an expression over symbols in \( T U E \):
    \[
    \alpha ::= S \mid empty \mid \alpha_1 | \alpha_2 \mid \alpha_1, \alpha_2 \mid \alpha? \mid \alpha^* \mid \alpha^+ \]
Basic Reasoning Tasks on XML DTDs—Conformance

- Definition of the set of document instances in terms of a DTD to which they conform to

**Definition 4** The set $\text{docs}(\mathbf{P}, S)$ of document instances generated by a set of element type definitions $\mathbf{P}$ starting from a symbol $S$ is inductively defined as follows:

- If $S$ is a terminal $F$, then $\text{docs}(\mathbf{P}, F) = F$.
- If $S$ is an element type $E$ and $E \rightarrow \alpha \in \mathbf{P}$, then $\text{docs}(\mathbf{P}, E)$ is the set of sequences $<E>d_1 \cdots d_k</E>$, where $<E>$ and $</E>$ are the start and end tags associated to $E$, and $d_1, \ldots, d_k$ are document instances generated by an instance of the content model $\alpha$. Formally:

$$\text{docs}(\mathbf{P}, E) = \{<E>d_1 \cdots d_k</E> \mid \text{there exists a word } S_1 \cdots S_k \text{ generated by } \alpha \text{ such that } d_i \in \text{docs}(\mathbf{P}, S_i) \text{, for } i \in \{1, \ldots, k\} \}$$

The set $\text{docs}(\mathbf{D})$ of document instances generated by a DTD $\mathbf{D} = (\mathbf{P}, \mathbf{R})$ is given by $\text{docs}(\mathbf{P}, R)$. A document instance $d$ conforms to a DTD $\mathbf{D}$ if $d \in \text{docs}(\mathbf{D})$.

Basic Reasoning Tasks on XML DTDs—Equivalence

- **Strong Equivalence (Inclusion, Disjointness):** the names of the tags are important.

**Definition 6** Given two DTDs $\mathbf{D}_1$ and $\mathbf{D}_2$,

- $\mathbf{D}_1$ is **strongly included** in $\mathbf{D}_2$, denoted with $\mathbf{D}_1 \sqsubseteq \mathbf{D}_2$, if $\text{docs}(\mathbf{D}_1) \subseteq \text{docs}(\mathbf{D}_2)$;
- $\mathbf{D}_1$ is **strongly equivalent** to $\mathbf{D}_2$, denoted with $\mathbf{D}_1 \equiv \mathbf{D}_2$, if $\text{docs}(\mathbf{D}_1) = \text{docs}(\mathbf{D}_2)$;
- $\mathbf{D}_1$ is **strongly disjoint** from $\mathbf{D}_2$, denoted with $\mathbf{D}_1 \ncong \mathbf{D}_2$, if $\text{docs}(\mathbf{D}_1) \cap \text{docs}(\mathbf{D}_2) = \emptyset$.

- **Structural Equivalence (Inclusion, Disjointness):** replace the tags into unnamed $<>$ or $</>$, and only consider the document structures.
Basic Reasoning Tasks on XML DTDs—Equivalence (cont’d)

Example of structural inclusion:

```
<!DOCTYPE Mail [<ELEMENT Mail (From, To, (Subject)?, Body)>]
<!ELEMENT From (Address)>  
<!ELEMENT To (Address)>    
<!ELEMENT Address (#PCDATA)>  
<!ELEMENT Subject (#PCDATA)>  
<!ELEMENT Body (#PCDATA | any)> ]>
```

DTD $M$ for mail documents

```
<!DOCTYPE Note [(From, To, Text)>]
<!ELEMENT From (Address)>  
<!ELEMENT To (Address)>    
<!ELEMENT Address (#PCDATA)>  
<!ELEMENT Text (#PCDATA | any)> ]>
```

DTD $N$ for note documents

---

Basic Reasoning Tasks on XML DTDs—Equivalence (cont’d)

- Natural generalization: $R$-equivalence (inclusion, disjointness):
  - View strong/structural equivalence (inclusion, disjointness) as extremes
  - $R$ is an equivalence relation on $E$
  - For each $E \in E$, denote $[E]_R$ the equivalence class of $E$ respect to $R$
    - e.g., in the previous example:
      - $[\text{Mail}]_R = \{\text{Mail, Note}\} = [\text{Note}]_R$
      - $[\text{Text}]_R = \{\text{Text, Body}\} = [\text{Body}]_R$
      - in which case $N \not\sim M$

11
Basic Reasoning Tasks on XML DTDs—Equivalence (cont’d)

• \( R \) -Conformance and \( R \) -equivalence (inclusion, disjointness):

\textbf{Definition 7} The set \( \text{docs}_R(P, S) \) of \( R \)-document instances generated by a set of element type definitions \( P \) starting from a symbol \( S \) is inductively defined as follows:

- If \( S \) is a terminal \( F \), then \( \text{docs}_R(P, F) = F \).
- If \( S \) is an element type \( E \) and \( (E \rightarrow \alpha) \in P \), then

\[
\text{docs}_R(P, E) = \{<E^i >d_1 \cdots d_k <E^i > | E^i \in [E]_R, \text{ and there exists a word } S_1 \cdots S_k \text{ generated by } \alpha \text{ such that } d_i \in \text{docs}_R(P, S_i), \text{ for } i \in \{1, \ldots, k\}\}
\]

The set \( \text{docs}_R(D) \) of \( R \)-document instances generated by a DTD \( D = (P, R) \) is given by \( \text{docs}_R(P, R) \). A document instance \( d \) is \( R \)-conforms to a DTD \( D \) if \( d \in \text{docs}_R(D) \).

\textbf{Definition 8} A DTD \( D_1 \) is \( R \)-included in a DTD \( D_2 \), denoted with \( D_1 \sqsubseteq_R D_2 \), if \( \text{docs}_R(D_1) \subseteq \text{docs}_R(D_2) \). \( R \)-equivalence, denoted with \( \equiv_R \), and \( R \)-disjointness, denoted with \( \ominus_R \), of two DTDs are defined in a similar way.

The Description Logic (DL) for Representing DTDs

<table>
<thead>
<tr>
<th>Concepts ( C )</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>atomic concept ( A )</td>
<td>( \Delta^2 \subseteq \Delta^2 )</td>
<td></td>
</tr>
<tr>
<td>top ( \top )</td>
<td>( \Delta^2 )</td>
<td></td>
</tr>
<tr>
<td>negation ( \neg C )</td>
<td>( \Delta^2 \setminus C^2 )</td>
<td></td>
</tr>
<tr>
<td>conjunction ( C_1 \land C_2 )</td>
<td>( C_1^2 \cap C_2^2 )</td>
<td></td>
</tr>
<tr>
<td>universal quantif. ( \forall R.C )</td>
<td>{( o \mid \forall o' \in R^2 \exists o' \in C^2 }}</td>
<td></td>
</tr>
<tr>
<td>qual. number restr. ( (\leq n)C )</td>
<td>{( o \mid \exists (o', o) \in P^2 \land o' \in C^2 \leq n }}</td>
<td></td>
</tr>
<tr>
<td>well-founded ( \text{wf}(R) )</td>
<td>{( o_0 \mid \forall o_1, o_2, \ldots (\text{well-founded}) \exists o \geq 0 : (o_0, o_{n+1}) \notin R^2 }}</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Roles ( R )</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>atomic role ( P )</td>
<td>( P^2 \subseteq \Delta^2 \setminus \Delta^2 )</td>
<td></td>
</tr>
<tr>
<td>union ( R_1 \cup R_2 )</td>
<td>( R_1^2 \cup R_2^2 )</td>
<td></td>
</tr>
<tr>
<td>conjunction ( R_1 \land R_2 )</td>
<td>( R_1^2 \cap R_2^2 )</td>
<td></td>
</tr>
<tr>
<td>refl. trans. closure ( R^+ )</td>
<td>( (R^2)^+ )</td>
<td></td>
</tr>
<tr>
<td>transitive closure ( \text{id}(C) )</td>
<td>{( (o, o) \mid o \in C^2 }}</td>
<td></td>
</tr>
</tbody>
</table>
The Description Logic (DL) for Representing DTDs (cont’d)

- Knowledge bases in DL:
  - A DL knowledge base \( K \) is a set of assertions of the form: \( C_1 \sqsubseteq C_2 \) or \( C_1 \equiv C_2 \), where \( C_1 \) and \( C_2 \) are arbitrary concepts with no restrictions
  - An interpretation \( I \) satisfies the \( C_1 \sqsubseteq C_2 \) if \( C_1 \models I \subseteq C_2 \models I \). \( I \) is a model of \( K \) if it satisfies all assertions in \( K \)
  - If \( I \) is considered a first order structure, checking whether \( I \) is a model of \( K \) has polynomial complexity

The Description Logic (DL) for Representing DTDs (cont’d)

- Reasoning services in DL:
  - Concept satisfiability: \( K \models C \sqsubseteq \bot \)
  - Knowledge base satisfiability: \( K \models \bot \)
  - Subsumption: \( K \models C_1 \sqsubseteq C_2 \) (can be reduced to \( K \models C_1 \equiv \bot \equiv C_2 \))
  - Theorem: It is sufficient to consider concept satisfiability only, which is an EXPTIME-complete problem
**Representation of DTDs in DL**

- Given a DTD \( D = (P, S_0) \), define a DL knowledge base \( K \), named characteristic knowledge base of \( D \).
- The alphabet of \( K \) (i.e., the atomic concepts and roles):
  - The atomic concepts \( \text{Tag} \) and \( \text{Terminal} \)
  - For each \( F \in T \), one atomic concept \( F \)
  - For each element type \( E \in E \), one atomic concept \( \text{Start}E \) and one atomic concept \( \text{End}E \)
  - The atomic roles \( f \) and \( r \) (standing for “first” and “rest” respectively)
  - For each element type definition, one atomic concept \( E_D \)
    --- All but atomic concepts \( E_D \) are basic atomic concepts/roles
  - The set of assertions of \( K \) is constituted by \( K_{TE} \), \( K_R \) and \( K_D \)

**Representation of DTDs in DL (cont’d)**

- General structure:
  - Example:
    - Tree representing the document with \( h \) internal components:
      \[
      \begin{aligned}
      &\text{<E>}\cr
      &\text{<E}_1\text{>...<E}_i\text{>}\cr
      &\text{<E}_2\text{>...<E}_j\text{>}\cr
      &\ldots\cr
      &\text{<E}_k\text{>...<E}_l\text{>}\cr
      &\text{</E>}
      \end{aligned}
      \]
  - Connection roles:
    - For start tag \( \text{<E>} \): \( f \)-filler
    - For end tag \( \text{</E>} \): \( r^{h+1} \)-filler
    - For the \( k \)th component:
      \[
      \begin{aligned}
      &\text{(<r^k f>)}-\text{filler}\cr
      &\text{(<1 \leq k \leq h>)}
      \end{aligned}
      \]
Representation of DTDs in DL (cont’d)

- $K_{T,E}$: Encoding of general structural properties.
  
- Assertions:
  
  $T \equiv (\leq 1 f, T) \cap (\leq 1 r, T) \cap \text{wf}(f \cup r)$
  
  $Tag \equiv \forall (f \cup r). \bot$
  
  Terminal $\equiv \forall (f \cup r). \bot \cap \neg Tag$
  
  $F \sqsubseteq \text{Terminal}$ for each terminal $F \in T$
  
  $F_1 \sqsubseteq \neg F_2$ for each pair of terminals $F_1, F_2 \in T$ such that $F_1 \neq F_2$
  
  $\text{Start}E \sqsubseteq \text{Tag}$ for each element type $E \in E$
  
  $EndE \sqsubseteq \text{Tag}$ for each element type $E \in E$

Representation of DTDs in DL (cont’d)

- $K_R$: Encoding of equivalence relation $R$
  
- Let $\{\{E_1^1, \ldots, E_1^n\}, \ldots, \{E_m^n, \ldots, E_m^n\}\}$ be the set of equivalence classes determined by $R$
  
- Assertions:
  
  $\begin{align*}
  \text{Start}E_i^j & \equiv \neg \text{Start}E_i^{j+1} \\
  \text{End}E_i^j & \equiv \neg \text{End}E_i^{j+1} \\
  \text{Start}E_i^n & \equiv \text{Start}E_i^{n+1} \\
  \text{End}E_i^n & \equiv \text{End}E_i^{n+1}
  \end{align*}$

  for $i, j \in \{1, \ldots, m\}$ and $i \neq j$

  for $i \in \{1, \ldots, n_j-1\}$ and $j \in \{1, \ldots, m\}$
Representation of DTDs in DL (cont’d)

- $\mathbf{K_D}$: Encoding of the DTD

Assertion: $E_D \equiv \exists f.\text{Start}\neg \exists (x \circ \tau(\alpha)).\text{End}\neg$

where, $\tau(\alpha)$: inductively defined complex role:

- $\tau(\text{empty}) = id(T)$
- $\tau(S) = id(\exists f.\text{ac}(D,S)) \circ r$
- $\tau(\alpha_1 \circ \alpha_2) = \tau(\alpha_1) \cup \tau(\alpha_2)$
- $\tau(\alpha_1, \alpha_2) = \tau(\alpha_1) \circ \tau(\alpha_2)$
- $\tau(\alpha^*) = \tau(\alpha)^*$
- $\tau(\alpha^+) = \tau(\alpha)^+$
- $\tau(\alpha?) = \tau(\alpha) \cup \text{id}(T)$

Atomic concept for the definition of a symbol:

$\text{ac}(D,S) = \{ E_D \text{ if } S = E \text{ for an element type } E \in E \}$

\[ \text{ if } S = F \text{ for a terminal } F \in T \]

Representation of DTDs in DL (cont’d)

- Example: $K_D$ for DTD $M$

```xml
<!DOCTYPE Mail [
  <ELEMENT Mail [From, To, (Subject)?, Body]>
  <ELEMENT Mail [From, To]>
  <ELEMENT From [Address]>
  <ELEMENT To [Address]>
  <ELEMENT Address [#PCDATA]>
  <ELEMENT Subject [#PCDATA]>
  <ELEMENT Body [#PCDATA | any]>
]>
```

Mail\(_M\) $\equiv$ ![StartMail] $\exists x \circ \text{id}(\exists f.\text{From}_M) \circ r \circ \text{id}(\exists f.\text{To}_M) \circ r$

From\(_M\) $\equiv$ ![StartFrom] $\exists x \circ \text{id}(\exists f.\text{Address}_M) \circ r \circ \text{id}(\exists f.\text{Body}_M) \circ r.$EndFrom

To\(_M\) $\equiv$ ![StartTo] $\exists x \circ \text{id}(\exists f.\text{Address}_M) \circ r \circ \text{id}(\exists f.\text{Body}_M) \circ r.$EndTo

Subject\(_M\) $\equiv$ ![StartSubject] $\exists x \circ \text{id}(\exists f.\text{Subject}_M) \circ r \circ \text{id}(\exists f.\text{Body}_M) \circ r.$EndSubject

Body\(_M\) $\equiv$ ![StartBody] $\exists x \circ \text{id}(\exists f.\text{Body}_M) \circ r.$EndBody

Address\(_M\) $\equiv$ ![StartAddress] $\exists x \circ \text{id}(\exists f.\text{Address}_M) \circ r.$EndAddress
Reasoning over DTDs in DL — $R$-Conformance

- There is a linear time construction of a one-to-one mapping $\beta$ from a $R$-document instance $d$ to a model of $K$.
- Example: Apply $\beta$ to the mail document instance conforming to DTD $M$.

Example:

```xml
<Mail>
  <From>
    <EmailAddress>David@Sun.com</EmailAddress>
  </From>
  <To>
    <EmailAddress>David@Citi.com</EmailAddress>
    <EmailAddress>Virgilio@Epge.mii</EmailAddress>
  </To>
  <Subject>Appointment</Subject>
  <Body>
    I can’t be there at four. Let’s do dinner at midnight.
    Tell also Carolee. Cheers,
    - J.J.
  </Body>
</Mail>
```

Reasoning over DTDs in DL — $R$-Conformance (cont’d)

- Theorem: verifying $R$-conformance of a document instance $d$ to a DTD $D$ can be polynomially reduced to model checking in DL.
  - The construction of $K$ takes polynomial time in the size of $D$.
  - The construction of a model based on a XML document instance $d$ takes linear time in size of $d$.
- Corollary: verifying $R$-conformance of a document instance $d$ to a DTD $D$ can be done in time polynomial in the size of $d$ and $D$. 
Reasoning over DTDs in DL—R-Equivalence (Inclusion, Disjointness)

- The checking of R-equivalence between two DTDs \( D \) and \( D' \) can be done by checking the equivalence between the atomic concepts for root element definitions under a unique model of \( K \), where \( K = K_{T,E} \cup K_{R} \cup K_{D} \cup K_{D'} \). Formally:

Theorem 19 Let \( D \) and \( D' \) be two DTDs, and \( K = K_{T,E} \cup K_{R} \cup K_{D} \cup K_{D'} \) be the knowledge base derived from \( T, E, R, (D, D') \) as described above. Then

\[
D \sqsubseteq_{R} D' \quad \text{if and only if} \quad K \vDash R_D \sqsubseteq R_{D'}
\]

(1)

\[
D \otimes_{R} D' \quad \text{if and only if} \quad K \vDash R_D \cap R_{D'} \sqsubseteq \perp
\]

(2)

Corollary 20 R-inclusion, R-equivalence, and R-disjointness between two DTDs can be verified in deterministic exponential time in the size of the DTDs.

Retrieving XML Documents from a Document Base

- Document base \( B = \langle D, I \rangle \)
  - \( D \): a set of DTDs. The relationship of each pair of DTDs is known
  - \( I \): a set of document instances
- Query \( Q \): a DTD query
- \( Q(B) \): the set of document instances in \( B \) R-conforming to \( Q \)
- Idea of the retrieval algorithm: choose the best DTD \( D \) from \( D \), which:
  - Satisfy \( D \models_{R} Q \), or
  - Satisfy \( D \not\models_{R} Q \), and there not exists \( D' \not\models_{R} Q \), and \( D \not\models_{R} D' \)
Conclusion and Discussion

What’s done:
- A view of DTDs as concepts of the expressive Description Logic DL is given
- Some questions regarding reasoning tasks on DTDs are answered (conformance, equivalence)
- Efficient algorithm for retrieving documents is proposed
- Some special aspects are taken into account:
  - Attributes as properties of elements
  - Documents containing links to other documents
  - etc.
  ---Ideas: take more advantage of existing DL (e.g., qualified number restrictions)
  introduce new concepts and roles

What’s to be explored:
- Compare the expressive power between the tree model given in this paper and other proposed data models (e.g., OEM, BDFS)
- Fully capture further aspects of DTDs
- Study new types of reasoning on DTDs
  - Parameterized equivalence
  - Document classification