IIC3432 - Tópicos Avanzados en Bases de Datos

Esquema de un documento XML

XML: Almacenando libros

```
<db>
   <book>
      <title> Algebra </title>
      <author>
         <name> Hungerford </name>
      </author>
   </book>
   <book>
      <title> Real Analysis </title>
      <author>
         <name> Royden </name>
      </author>
   </book>
</db>
```

XML: Almacenando libros

- No todos los documentos XML almacenan información sobre libros:

```
<email>
   <from> marenas@ing.puc.cl </from>
   <to> jperez@ing.puc.cl </to>
   <date> 10/4/200 </date>
   <subject> Recordatorio </subject>
</email>
```

- ¿Cómo especificamos la estructura de una clase de documentos XML?

Esquema de un documento: DTD

Un ejemplo:

```
<!DOCTYPE db [
   <!ELEMENT db (book*)>
   <!ELEMENT book (title, author*)>
   <!ELEMENT title #PCDATA>
   <!ELEMENT author #PCDATA>
]>
```

Esquema de un documento: DTD

Dado: un conjunto E de tipos.

- Ejemplo: $E = \{db,\ book,\ title,\ author,\ name\}$.

DTD d: Para cada elemento e se tiene una regla de la forma:

<!ELEMENT e r >

donde r es:

- EMPTY
- ANY
- o una expresión regular sobre el alfabeto $E \cup \{\texttt{\#PCDATA}\}$.

Esquema de un documento: DTD

- Expresiones regulares pueden incluir `?` y `+`:

```
<!ELEMENT db (book+)>
<!ELEMENT db (title?, author+)>
```

$r? = r|\varepsilon$ y $r^+ = r, r^*$.

- Raíz del documento: `<!DOCTYPE db [ ... ]>`

DTD: Formalización

Definición: Un DTD d sobre un conjunto E de tipos es una tupla (e, p):

- $e \in E$ es la raíz.
- $p : E \to 2^{(E \cup \{\texttt{\#PCDATA}\})^*}$ envía cada tipo en E a una expresión regular sobre $E \cup \{\texttt{\#PCDATA}\}$.

Observación: Usamos ε para `EMPTY` y $(E \cup \{\texttt{\#PCDATA}\})^*$ para `ANY`.

Un E-árbol $T = (D, \lambda)$ satisface d, denotado como $T \models d$, si para cada $s \in D$:

- s es hoja: $\varepsilon \in L(p(\lambda(s)))$.
- s tiene hijos $s \cdot 0, \ldots, s \cdot n$: $\lambda(s \cdot 0) \cdots \lambda(s \cdot n) \in L(p(\lambda(s)))$.

DTDs recursivos

Un DTD puede ser recursivo:

```
<!ELEMENT node (leaf | (node,node))>
```

Ejercicio: Construya un DTD que defina la clase de los árboles binarios que representan traducciones de árboles con una cantidad arbitraria de hijos con etiquetas a y b, y que sólo pueden tener a como raíz.

¿Qué problema podemos tener con los DTDs recursivos?

```
<!ELEMENT nodo (nodo+)>
```

DTDs recursivos: Consistencia

Decimos que un DTD d es consistente si existe un árbol T tal que $T \models d$.

¿Cómo podemos determinar si un DTD d es consistente?

DTD: Desde la práctica a la teoría

Lo siguiente aparece en Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 04 February 2004 (`http://www.w3.org/TR/REC-xml/`):

> As noted in 3.2.1 Element Content, it is required that content models in element type declarations be deterministic. This requirement is for compatibility with SGML (which calls deterministic content models "unambiguous"); XML processors built using SGML systems may flag non-deterministic content models as errors.

DTD: Desde la práctica a la teoría

For example, the content model ((b, c) | (b, d)) is non-deterministic, because given an initial b the XML processor cannot know which b in the model is being matched without looking ahead to see which element follows the b. In this case, the two references to b can be collapsed into a single reference, making the model read (b, (c | d)). An initial b now clearly matches only a single name in the content model. The processor doesn't need to look ahead to see what follows; either c or d would be accepted.

DTD: Desde la práctica a la teoría

¿Qué preguntas dejan planteadas los párrafos anteriores?

> As noted in 3.2.1 Element Content, it is required that content models in element type declarations be deterministic. ...
> For example, the content model ((b, c) | (b, d)) is non-deterministic, because given an initial b the XML processor cannot know which b in the model is being matched without looking ahead to see which element follows the b ...

¿Qué significa que una expresión regular sea determinista?

DTD: Desde la práctica a la teoría

... XML processors built using SGML systems may flag non-deterministic content models as errors.

¿Cómo podemos verificar si una expresión regular es determinista?

... In this case, the two references to b can be collapsed into a single reference, making the model read (b, (c | d)) ...

¿Una expresión regular es siempre equivalente a una expresión regular determinista?