

14 de septiembre 2006

1 Escribir el XML Schema correspondiente

```
| - xs:schema
| - xs:element (name="restaurante")
| - xs:complexType
| - xs:sequence
| - xs:element (name="mesa",maxOccurs="unbounded")
| - xs:complexType
| - xs:sequence
| - xs:element (name="numero",type="xs:integer")
| - xs:element (name="fecha",type="xs:date")
| - xs:element (name="hora",type="xs:time")
| - xs:element (name="camarero",type="xs:string")
| - xs:element (name="plato")
| - xs:complexType
| - xs:sequence
| | - xs:element (name="nombre",type="xs:string")
| | - xs:element (name="precio")
| | | - xs:simpleType
| | | - xs:restriction (base="xs:integer")
| | | - xs:minInclusive (value="1")
| | | - xs:maxInclusive (value="100000")
| | - xs:element (name="descuento")
| | - xs:simpleType
| | - xs:restriction (base="xs:decimal")
```

```
| - xs:minInclusive (value="1")
| - xs:maxInclusive (value="100")
| - xs:fractionDigits (value="1")
| - xs:attribute (name="id",type="xs:string")
```

2 Escribir las siguiente dependencias funcionales

1. restaurante.mesa \rightarrow restaurante.mesa.camarero
2. restaurante.mesa.plato.nombre.# \rightarrow restaurante.mesa.plato.precio.#
3. restaurante.mesa \rightarrow restaurante.mesa.descuento.#

3 XNF

1. restaurante.mesa \rightarrow restaurante.mesa.numero
2. Primero, examinamos la validez de cada DF:
 - (a) Es valida (la parte derecha es un camino "elemento")
 - (b) No es valida: usamos el contre-ejemplo

```

<restaurante><mesa>...
<plato><nombre>pollo
asado</nombre><precio>1</precio>...</plato>
<plato><nombre>pollo
asado</nombre><precio>1</precio>...</plato>
...</mesa>...</restaurante>

```

Tenemos dos tuplas t_1 y t_2 tal cual $t_1(p) = t_2(p)$ por los caminos p “restaurante.mesa.plato.nombre.#” y “restaurante.mesa.plato.precio.#” pero que son diferentes por el camino “restaurante.mesa.plato.precio”

(c) No es valida: usamos el contre-ejemplo

```

<restaurante><mesa>...
<plato>...<descuento>1</descuento>...</plato>
<plato>...<descuento>1</descuento>...</plato>
...</mesa>...</restaurante>

```

Tenemos dos tuplas t_1 y t_2 tal cual $t_1(p) = t_2(p)$ por los caminos p “restaurante.mesa” y “restaurante.mesa.plato.descuento.#” pero que son diferentes por el camino “restaurante.mesa.plato.descuento”

Para corregir el XML schema: (1) El elemento “descuento” debe ser un hijo (unico) de “mesa” y no de “plato” (2) El elemento “plato” debe ser un hijo de “restaurante” y el plato dentro de la mesa debe tener una referencia hacia este elemento.

Para lograr eso, modificamos el XML schema de la manera siguiente: desplazamos la definición de descuento despues de la definición del elemento “numero”. Desplazamos la definición del elemento “plato” antes de la definición del elemento “mesa” y en vez de la antigua definición (dentro de “mesa”), ponemos: `<xs:element name='plato' type='xs:string'/>` que hace referencia al id del elemento plato.

4 Ampliar el XML Schema

4.1 XML Schema

```

<xs:element name="pago"><xs:complexType><xs:sequence>
  <xs:element name="tarjeta" type="xs:integer">
    <xs:choice>
      <xs:element name="tarjeta" type="xs:integer">
      <xs:element name="efectivo" type="xs:integer">
    </xs:choice>
  </xs:sequence></xs:element></xs:element>

```

4.2 Interpretación

La gramatica formal es $D = (N, T, P, S)$:

$$N = \{\text{Pago, Tarjeta, Efectivo}\}$$

$$T = \{\text{pago, tarjeta, efectivo}\}$$

$$S = \{\text{Pago}\}$$

P es compuesto de las reglas siguiente: $p_1 = \text{Pago} \rightarrow \text{pago}$ (Tarjeta*, (Tarjeta|Efectivo)), $p_2 = \text{Tarjeta} \rightarrow \text{tarjeta } \epsilon$, $p_3 = \text{Efectivo} \rightarrow \text{efectivo } \epsilon$

Usando una notación formal por el documento XML: $V = \{a, b\}$
 $lab(a) = \text{pago}$ $lab(b) = \text{tarjeta}$ $ch(a) = \{b\}$ $ch(b) = \emptyset$

Eso es una interpretación valida porque existe una interpretación valida $I : V \rightarrow P$:

- $I(a) = p_1$, $lab(a) = non - terminal(I(a))$, $I(a) \in S$ y la secuencia (Tarjeta) de los non-terminales asociada a la secuencia de los hijos puede ser generada por la expression regular (Tarjeta*, (Tarjeta|Efectivo))
- $I(b) = p_2$, $lab(b) = non - terminal(I(b))$ y la secuencia vacia corresponde a la secuencia vacia de los hijos.