

<pre> <a> <e>ciao</e> <f>20</f> <c> <h>20</h> <g>ciao</g> <i>38</i> <i>98</i> </c> </pre>	<pre> <a> <e>ciao</e> <z>20</z> <z>30</z> <z>40</z> <e>ciao</e> <d>maramao</d> <c> <m>20</m> <g>ciao</g> </c> </pre>
---	---

Figura 1 Due documenti XML

Esercizio

Si considerino i due documenti XML in Figura. Scrivere un DTD e un XML schema che generalizzi la struttura dei due documenti. Il DTD/schema deve descrivere i due documenti in modo conciso. I due documenti devono essere validi per tale DTD/schema.

Soluzione

L'esercizio richiede di generare un DTD/schema a partire dai due documenti. Quindi i due documenti devono risultare validi rispetto allo schema/DTD generato.

Di seguito viene presentato un DTD per tali documenti:

```

<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XMLSpy v2005 sp1 U (http://www.xmlspy.com)-->
<!ELEMENT a (b+, d?, c)>
<!ELEMENT b (e, f?, z*)>
<!ELEMENT e (#PCDATA)>
<!ELEMENT f (#PCDATA)>
<!ELEMENT z (#PCDATA)>
<!ELEMENT d (#PCDATA)>
<!ELEMENT c ((h | m), g, i*)>
<!ELEMENT h (#PCDATA)>
<!ELEMENT m (#PCDATA)>
<!ELEMENT g (#PCDATA)>
<!ELEMENT i (#PCDATA)>

```

Di seguito viene presentato un XML schema per tale documento. Si noti che nello schema non vengono introdotti elementi globali (al di fuori della radice) perché nessun elemento viene utilizzato in contesti diversi (o in altre parole un elemento non è mai sottoelemento di due elementi distinti).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="a">
<xs:complexType>
<xs:sequence>
  <xs:element name="b" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
  <xs:element name="e" type="xs:string"/>
  <xs:element name="f" type="xs:integer" minOccurs="0"/>
  <xs:element name="z" type="xs:integer" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="d" type="xs:string" minOccurs="0"/>
<xs:element name="c">
<xs:complexType>
<xs:sequence>
  <xs:choice>
    <xs:element name="h" type="xs:integer"/>
    <xs:element name="m" type="xs:integer"/>
  </xs:choice>
  <xs:element name="g" type="xs:string"/>
  <xs:element name="i" type="xs:integer" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```