

INFO-H-509

Exercises 2

XML Schema

XML Schema

Cardinality

```
<xs:element name="a"  
  minOccurs="4"  
  maxOccurs="6"  
/>
```

- Element a must appear between 4 and 6 times.

```
<xs:sequence  
  minOccurs="0"  
  maxOccurs="unbounded">  
  ...  
</xs:sequence>
```

- The whole sequence is optional. The number of occurrences is however not restricted.

XML Schema

Sequence

```
<xs:sequence>  
  <xs:element name="a" />  
  <xs:element name="b" />  
  <xs:element name="c" />  
</xs:sequence>
```

- a, b, and c in this precise order

XML Schema

Choice

```
<xs:choice>  
  <xs:element name="a" />  
  <xs:element name="b" />  
  <xs:element name="c" />  
</xs:choice>
```

- One of a, b, or c

XML Schema

All

```
<xs:all>  
  <xs:element name="a" />  
  <xs:element name="b" />  
  <xs:element name="c" />  
</xs:all>
```

- a, b, and c, in any order

maxOccurs must be ≤ 1

DTD

Binding XML to DTD

```
<!DOCTYPE {root-element}
  SYSTEM '{uri}' [
  {definitions}
]>
```

Element

```
<!ELEMENT {name} {content-model}>
```

Content models:

(#PCDATA {e1} {e2} ...)	Mixed
EMPTY	
ANY	
{e1}, {e2}, {e3}, ...)	Sequence
{e1} {e2} {e3} ...)	Choice

Cardinality

?	0-1
*	0-inf
+	1-inf

Elements of sequence and choice can in turn be sequences or choices, with cardinality specifiers.

Mixed can be reduced to (#PCDATA) to only accept text.

Attribute list

```
<!ATTLIST {element}
  {att1} {type1} {opt1}
  {att2} {type2} {opt2} {def2}
  ...
  {attn} {typen} {optn}
>
```

Type

CDATA	Any text
{v1} {v2} {v3}	List of values
NMTOKEN, NMTOKENS	(List of) XML names
ID	Unique identifier
IDREF	Reference to an ID
ENTITY, ENTITIES	

Options

#REQUIRED	
#IMPLIED	Optional
#FIXED	Cannot be changed

Exercices

- Validation tools

```
java -jar DTDValidation.jar <xml doc>
```

```
java -jar XSDValidation.jar <schema> <xml doc>
```

Deterministic Regular Expressions

1. For a regular expression a , define a' to be the regular expression obtained by replacing the i -th occurrence of symbol s by s_i
 - $a = (a \mid b)^+ cba^* (a \mid c)$
 - $a' = (a_1 \mid b_1)^+ c_1 b_2 a_2^* (a_3 \mid c_2)$
2. The regular expression is deterministic if there are no two strings $wb_i v$ and $wb_j z$ ($i \neq j$) in the regular language
 - Consider $a_1 c_1 b_2 a_2 a_3$ and $a_1 c_1 b_2 a_3 c_2$