

# Rapport XML

## Validation XSD :

- Le simple Type “cell” assure grâce à “xsd:restriction” que le type de chaque case un entier compris entre 1 et 9.

```
<xsd:simpleType>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1" />
    <xsd:maxInclusive value="9" />
  </xsd:restriction>
</xsd:simpleType>
```

- L'élément sudoku est un type complexe comprenant exactement 9 éléments “region” eux même de type complexe. Ces régions sont composés de séquences d'élément “row” et d'un attribut name, ces “rows” sont eux même composés d'éléments de types complexe “col” et d'un attribut “name”. On arrive alors au “cell” défini auparavant qui ont pour attribut “name”, une chaîne de caractère et “val”, la valeur de la cellule, un entier. Le nombre de régions est fixé à 9 et chaque région est composée de 9 case soit 3 lignes de 3 colonnes.

```
<xsd:element name="sudoku">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="region" minOccurs="9" maxOccurs="9">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="row" minOccurs="3" maxOccurs="3">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="col" minOccurs="3" maxOccurs="3">
                    <xsd:complexType>
                      <xsd:simpleContent>
                        <xsd:extension base="cell">
                          <xsd:attribute name="name" type="xsd:string" />
                          <xsd:attribute name="val" type="xsd:integer" />
                        </xsd:extension>
                      </xsd:simpleContent>
                    </xsd:complexType>
                  </xsd:element>
                </xsd:sequence>
                <xsd:attribute name="name" type="xsd:string" />
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
          <xsd:attribute name="name" type="xsd:string" />
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

## Validation XSLT :

- Le document XSL donne d'abord l'aperçu de la grille de Sudoku en SVG en parcourant chaque colonnes de chaque ligne de chaque région. Les variables xLoc et yLoc permettent de donner la position de chaque case de façon automatique. Il en est de même pour le texte. Le "-1" dans l'attribut des variables permet de décaler les valeurs, nous nous sommes rendu compte que lorsque nous avons fait les fichiers XML, nous avons compté les régions, les lignes et les colonnes à partir de 1 ce qui décalait les valeurs dans le fichier. En ajoutant ce -1, les valeurs sont correctement affichées dans la grille.

```
11         <xsl:for-each select="row">
12             <xsl:variable name="xLoc" select="substring(@name,4,4)-1" as="xsd:integer()" />
13             <xsl:for-each select="col">
14                 <xsl:variable name="yLoc" select="substring(@name,4,4)-1" as="xsd:integer()"
/>
```

- Voici les fichiers SVG obtenu :  
Pour le sudoku gagnant :

8	9	4	6	7	3	2	5	1
1	5	7	2	9	8	3	4	6
3	6	2	4	5	1	8	9	7
9	8	3	7	6	4	1	2	5
2	4	6	1	3	5	7	8	9
5	7	1	9	8	2	4	6	3
7	3	8	5	4	9	6	1	2
4	1	9	3	2	6	5	7	8
6	2	5	8	1	7	9	3	4

Pour le sudoku correct :

5	6		8	4	7			
3		9				6	6	
		8						
	1			8			4	
7	9		6		2		1	8
	5			3			9	
						2		
		6				8		7
			3	1	6		5	9

Pour le sudoku incorrect :

5	6		8	4	7			
3		9				6	6	
		8						
	1			8			4	
7	9		6		2		1	8
	5			3			9	
						2		
		6				8		7
			3	1	6	7	5	9

- On teste ensuite le type de grille (gagnante, correct ou incorrect).
- L'idée de base est la suivante :
  - On commence par regarder si la grille est gagnante donc pleine avec seulement une occurrence des chiffres de 1 à 9 dans une même région, une même ligne et une même colonne. On regarde d'abord

chaque colonne pour voir si le même chiffre est présent plusieurs fois. Si la valeur n'apparaît qu'une seule fois alors on

- Ensuite, si la grille n'est pas remplie, on regarde si elle est correcte ou non. Pour cela, on parcourt de nouveau chaque colonne de chaque ligne de chaque région. S'il n'y a pas de conflit sur une ligne, colonne ou région alors la grille est correcte sinon elle est incorrecte.

Les fichiers "sudoku\_correct.xml", "sudoku\_gagnante.xml", "sudoku\_incorrect.xml", et "sudoku\_incorrect2.xml" devraient permettre de tester le code XSL afin de connaître le statut de la grille. Toutefois, nous n'avons pas réussi à valider les grilles ou non, le résultat du test ne s'affiche pas dans le SVG.