

Semantic Web

Assignment 3

Dr. Jandson S Ribeiro

jandson@uni-koblenz.de

Isabelle Kuhlmann

iskuhlmann@uni-koblenz.de

Institute of Web Science and Technologies

Department of Computer Science

University of Koblenz-Landau

Submission by: May 16, 2021

Tutorial on: May 20, 2021

Please submit your solutions to your group's OLAT folder.
Always list all group members contributing to the solution!
Do not plagiarize from others!

For all the assignment questions that require you to code, make sure to include the code in the answer sheet, along with a separate Python file.


Team Name: gamma

1. Aman Bisht (amanvista@uni-koblenz.de)
2. Muralikrishna Naripeddi (mnaripeddi@uni-koblenz.de)
3. Ndang Hesley Fonane (fonaneh@uni-koblenz.de)



1 XML Validation and XML Schema

2 points



Consider the following XML document:

```
1: ?xml version="1.1" encoding="UTF-8" ?>
2: <!DOCTYPE book [
3:     <!ELEMENT book (title | author | publisher | year) >
4:     <!ELEMENT title (#PCDATA)>
5:     <!ELEMENT author (#PCDATA) >
6:     <!ELEMENT publisher (#PCDATA)>
7:     <!ELEMENT year (EMPTY)>
8: ]>
9: <book>
10:     <author><publisher>Macmillan</author></publisher>
11:     <title>Alice Adventures in Wonderland</title>
12:     <year>1865</year>
13: </book>
```

1.1

Is the XML document above valid or not (according to its DTD). Provide arguments for your answer. If it is invalid, point out at least 3 issues.

It is not valid according to its DTD.

XML Issues:

- line 1 should start with '<'
- author and publisher tags are not properly nested

DTD Issues:

- Element author should not have any sub elements as per the DTD, but author is having publisher tag as its child.
- In line number 7 the syntax for defining an EMPTY element is incorrect, instead it should be <!ELEMENT year EMPTY>
- Year element is defined EMPTY in DTD, but the content in year tag is 1865, which is not empty.
- As per the DTD only one of the children of the book tag should exist. From title, author, publisher, year only one of them should exist, but in the xml there are author, publisher, title, year are existing simultaneously.

2 XML-DTD


8 points

Consider the following movie domain specification.

A director, who has a first name and a last name, directs multiple films. A film has a single title, a publishing year, a studio, and it might be based on a book. A book has a title as well, and it has one or multiple authors.


2.1

2 points

Write a DTD document for the movie domain above. The outermost class must be  director. You must submit your DTD in a separate file.


2.2

2 points

Translate your DTD document for the movie domain above into a XML Schema . The outermost class must be director. You must specify at least one mandatory attribute, and at least two entities. You must submit your XML Schema in a separate file.

2.3

4 points

Write an XML document that conforms with the XML Schema you created in the  previous task. Your XML document must encode the following information:

David Fincher is an American director. Famous examples of his work are *Se7en* (production studio: New Line Cinema) from 1995 and *Fight Club* from 1999 (20th Century Fox). Although the movie *Fight Club* is well-known around the globe, the fact that it is actually based on a novel under the same name, written by Chuck Palahniuk, is lesser known.

3 Programming

10 Points

For this task, you shall write some code in Python that can access the XML file you created in the previous task. You must submit your code in a single .py file named **task3.py**. You can use the lxml library (<https://lxml.de>) for this task. The program must do the following:

3.1

**4 point**

Create a function with the following signature `validadeXMLFile(XMLfile, schemaFile)`. The function must validate the XM file against the schemaFile, and print: *“File Validated”*, if the XML file conforms with its DTD or XML Schema; otherwise, it must print *“Your XML file does not conform with its DTD/Schema”*. For the purpose of this task, assume that schemaFile is either an XML schema or DTD file. For this, you must verify the extension of the file: the extension for DTD is .dtd, and for XML schema the extension is .xsd. **Your function must follow the signature given above. Submit your code in the task3.py file.**

3.2

3 points

Create a function with the following signature: `parseInDepth(XMLFile)`. The function must visit each node of the XMLfile document top-down in a depth-first policy, and return a list of these nodes in the visiting order. **Your function must follow the given signature above. Submit your code in the task3.py file.**

3.3

3 points

Create a function with the following signature: `printXMLFile(XMLFile)`. The function must print in a human readable representation (e.g. an indented list) the XMLFile. You may use the function `parseInDepth` of the previous task to help you in this task. **Your function must follow the given signature above. Submit your code in the task3.py file.**

Important Notes

Submission

- Solutions have to be submitted to your group's OLAT folder.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the OLAT folder. Upload *all* `.py` files of your program! Use **UTF-8** as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
 - Make sure you code has consistent [indentation](#).
 - Make sure you comment and document your code adequately in English.
 - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This pdfLaTeX template was adapted by Isabelle Kuhlmann based on the LuaLaTeX version by Lukas Schmelzeisen.

ℒTeX

Use `pdflatex assignment_X.tex` to build your PDF.