Shiraz University
Advanced Databases Systems

Assignment 3

XML Validation and Querying

Department of Electrical and Computer Engineering

Instructor: Professor Sadreddini

Deadline:

22 Azar 1404, at 11:59 PM (Tehran Time)

Introduction

XML is a widely used markup language designed for storing and exchanging structured information in a form that is both human-readable and machine-readable. Ensuring the correctness and validity of an XML document typically requires using validation mechanisms such as **DTD** and **XML Schema**. Each provides a different way to define the allowed structure and datatypes of an XML file.

In this assignment, you will work with a real XML dataset from the web:

- `plant_catalog.xml` (W3Schools; public dataset)

You will design both a DTD and an XML Schema for this dataset, and you will also write several XPath expressions to query its hierarchical structure.

Your final submission must be a single ZIP file containing all requested XML, DTD, XSD, XPath, and PDF report files.

Part 1 – XML Validation (DTD & XML Schema)

In this part, you must write a DTD and an XML Schema for the file `plant_catalog.xml`. The dataset contains a root element `CATALOG`, and each plant is represented by a `PLANT` element with the following children:

- `COMMON`
- `BOTANICAL`
- `ZONE`
- `LIGHT`
- `PRICE`
- `AVAILABILITY`

You may download the dataset from:

`https://www.w3schools.com/xml/plant_catalog.xml`

1.1 DTD Design

Create a DTD file named `plants.dtd` that satisfies the following requirements:

(a) Every `PLANT` must contain exactly one occurrence of each of the elements listed above.

(b) The `ZONE` value must represent a natural number (e.g. 1–9).

(c) The `LIGHT` element may contain only one of the allowed values such as:

`Shade`, `Mostly Shady`, `Sun or Shade`, `Mostly Sunny`, `Sunny`

(d) The `PRICE` element must contain a decimal value. For the DTD, you may approximate the datatype using a general pattern or plain #PCDATA; specify the limitation in comments if needed.

(e) The `AVAILABILITY` date should follow the pattern `MM-DD-YYYY`. You may approximate this constraint in the DTD but should state its limitations.

(f) Each `PLANT` must have an attribute `code`:

- It must be unique.
- It must begin with the letter "p" followed by one or more digits (e.g. p10, p202).

(g) Include the proper DOCTYPE declarations in two XML files:

- `xml.plant_catalog_dtd.xml`: referencing an external DTD.
- `xml.plant_catalog_embed_dtd.xml`: containing an embedded/internal DTD.

Note: Some constraints cannot be fully enforced using DTD. Clearly identify these limitations in your report.

1.2 XML Schema (XSD)

Create an XML Schema file named `plants.xsd` that enforces all structural and datatype constraints more strictly. The Schema must satisfy the following:

(a) Use precise datatypes:

- `ZONE`: `xs:integer` with a value range (e.g. 1 to 9)
- `LIGHT`: `xs:string` with enumerated allowed values
- `PRICE`: `xs:decimal` with a minimum of 2 and maximum of 25
- `AVAILABILITY`: use `xs:string` with a date pattern, or an `xs:date` with restrictions via `pattern`

(b) Attribute `code` must match the pattern `p[0-9]+` and must be unique across all `PLANT` elements (use `xs:unique` or `xs:key`).

(c) You must use at least one `xs:key / xs:keyref` pair (for any additional referencing structure you design or extend).

(d) Create a separate XML file named `xml.plant_catalog_xsd.xml` that properly references the Schema using `xsi:schemaLocation`.

1.3 Documentation in Report

Your report must briefly explain:

- Which constraints could *not* be enforced in the DTD and why.
- How the XML Schema fully enforces these constraints.
- What tools or online validators you used to test your DTD and Schema.

You may use XML validation tools or editor plugins (such as XML Tools for Notepad++, Visual Studio Code XML extensions, or online DTD/XSD validators) to verify the correctness of your DTD, XML Schema, and to test the output of your XPath expressions.

Part 2 – XPath Queries

XPath is a tree-navigation language used to extract elements and attributes from an XML document. Based on the structure of `plant_catalog.xml`, write XPath expressions for each of the following tasks.

X2-1. Return the `COMMON` names of all plants whose `PRICE` value starts with the substring "`$2.`" (for example, `$2.44`, `$2.80`, etc.).

X2-2. Return the `BOTANICAL` names of all plants whose `LIGHT` value contains the substring "Shade".

X2-3. Return the `COMMON` names of plants whose `ZONE` value (as text) is exactly "3" or exactly "4".

X2-4. Return the `BOTANICAL` names of plants whose `LIGHT` value is exactly "Sunny" and whose `PRICE` value starts with the substring "`$9.`".

X2-5. Return the `COMMON` names of plants that satisfy all of the following: their `PRICE` value contains the digit "8" immediately after "`$`" (for example, `$8.60`, `$8.86`, `$8.56`), their `LIGHT` value contains the substring "Sun", and their `ZONE` value (as text) is either "4" or "5".

All XPath expressions must be placed in a text file named `xpath_queries.txt`.

Submission Instructions

1. Your ZIP file must contain the following:
   - `plants.dtd`
   - `xml.plant_catalog_dtd.xml`
   - `xml.plant_catalog_embed_dtd.xml`
   - `plants.xsd`
   - `xml.plant_catalog_xsd.xml`
   - `xpath_queries.txt`
   - `report.pdf`
2. Submit a single zipped archive (`.zip only; not .rar`) by email to: `adb.graders@gmail.com`. Use the following exact format:
   - Archive Name: `ADB-HW3-Jack_White-90012768.zip`
   - Email Subject: `Assignment 3:  90012768`

   Replace "Jack White" with your full name and "90012768" with your student number.
3. All deliverables must be included inside this single compressed archive file.
4. Late submissions will not be accepted unless explicitly approved by the instructor.