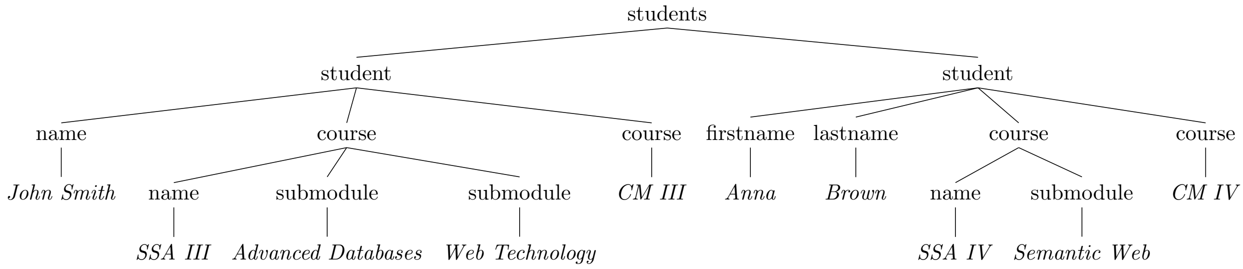
Advanced Databases

ffgt86

February 12, 2020

Part A

a) *Draw the directed tree structure of* `students.xml`.



b) *Write a DTD for* teachers.xml.

This DTD was validated using the `lxml` Python package. The provided file, `teachers.xml`, was modified slightly to facilitate this.

```
<?xml encoding="UTF-8"?>
<!ELEMENT teachers (teacher*)>
    <!ELEMENT teacher (name, course*)>
    <!ATTLIST teacher
        jobRole CDATA #REQUIRED
        joiningDate CDATA #REQUIRED>
    <!ELEMENT course (name,submodule+)>
        <!ELEMENT submodule (name,year+)>
            <!ELEMENT year (#PCDATA)>
            <!ELEMENT name (#PCDATA)>
```

However, with limited examples, and no formal specification, there are ambiguities:

- Should a teacher teach at least one course to be included in `teachers.xml`, i.e. `<!ELEMENT teacher (name, course+)>`?

- Should a course contain at least one submodule to be included in `teachers.xml`, i.e. `<!ELEMENT course (name, submodule+)>`?
- Which attributes (if any) are `#REQUIRED`, `#IMPLIED`, or `#FIXED`? Are there default values?
- Are there are limited number of values for `jobRole`? If there were only two roles available, for instance, `Professor` and `Researcher`, the attribute declaration should be `<!ATTLIST teacher jobRole(Professor | Researcher)>`.

c) *Is it possible to write a DTD for* `students.xml`?

It is not possible. The difficult construct is `<course>`. A student needs one or more courses (`course+`) but a course can either be `#PCDATA` or `(name, submodule*)`, and DTD does not allow mixed delcarations such as `<!ELEMENT course (#PCDATA | (name, submodule*)>`. This problem could be eliminated by forbidding `#PCDATA` in `<course>` and always using the `<name>` element, i.e. by changing:

```
<student enrolmentDate="2016">
        <firstname>Anna</firstname>
        <lastname>Brown</lastname>
        <course>
            <name>Software, Systems and Applications IV</name>
            <submodule>Semantic Web</submodule>
        </course>
        <course>Computing Methodologies IV</course>
    </student>
```

to:

```
<student enrolmentDate="2016">
        <firstname>Anna</firstname>
        <lastname>Brown</lastname>
        <course>
            <name>Software, Systems and Applications IV</name>
            <submodule>Semantic Web</submodule>
        </course>
        <course>
            <name>Computing Methodologies IV</name>
        </course>
    </student>
```

The problematic DTD element would then be `<!ELEMENT course (name, submodule*)>`

Part B

a) *Find all students who study "Advanced Databases" this year.*

1. Select students for this year:

```
/students[@year='2019-2020']
```

2. Select all branches with a `submodule` node with a text value of 'Advanced Databases':

```
/student/course/submodule[text()='Advanced Databases']
```

3. Return the `<student>` ancestors from the matches in step 2:

```
/ancestor::student
```

The output is:

```
<student enrolmentDate="2017">
       <name>John Smith</name>
       <course>
           <name>Software, Systems and Applications III</name>
           <submodule>Advanced Databases</submodule>
           <submodule>Web Technology</submodule>
       </course>
       <course>Computing Methodologies III</course>
   </student>
```

This approach is robust; there are no obvious limitations.

b) *Find all teachers who teach "Advanced Databases" this year.*

1. Select all `submodule` branches with nodes `name` and `year` with values 'Advanced Databases' and '2019-2020', respectively:

```
/teachers/teacher/course/submodule[name='Advanced Databases' and year='2019-2020']
```

2. Return the `<teacher>` ancestors from the matches in step 1:

```
/ancestor::teachers
```

The output is:

```
<teacher joiningDate="2018" jobRole="Professor">
        <name>Alexandra Cristea</name>
        <course>
            <name>Software, Systems and Applications III</name>
            <submodule>
                <name>Advanced Databases</name>
                <year>2018-2019</year>
                <year>2019-2020</year>
            </submodule>
        </course>
        <course>
            <name>Software, Systems and Applications IV</name>
            <submodule>
                <name>Semantic Web</name>
                <year>2018-2019</year>
                <year>2019-2020</year>
            </submodule>
        </course>
    </teacher>
```

This approach is robust; there are no obvious limitations.

c) *How many years has Professor Cristea been teaching "Advanced Databases" (at Durham)?*

XQUERY

d) *Find all students in year 3 currently taught by Alexandra.*

XQUERY

e) *How many teachers and how many students are kept in the databases where the last name is not known?*

XQUERY