```
      <element name="email" type="string"
        minOccurs="0" maxOccurs="1"/>
    </sequence>
    <attribute name="title" type="string" use="optional"/>
    <attribute name="rank" type="string" use="required"/>
  </complexType>
```

A hierarchical relationship exists between the original and the extended type. *Instances of the extended type are also instances of the original type*. They may contain additional information, but neither less information nor information of the wrong type.

Data Type Restriction

An existing data type may also be restricted by adding constraints on certain values. For example, new type and use attributes may be added, or the numerical constraints of minOccurs and maxOccurs tightened.

It is important to understand that restriction is *not* the opposite process of extension. Restriction is not achieved by deleting elements or attributes. Therefore, the following hierarchical relationship still holds: *Instances of the restricted type are also instances of the original type*. They satisfy at least the constraints of the original type and some new ones.

As an example, we restrict the lecturer data type as follows:

```
  <complexType name="restrictedLecturerType">
    <complexContent>
      <restriction base="lecturerType">
        <sequence>
          <element name="firstname" type="string"
            minOccurs="1" maxOccurs="2"/>
        </sequence>
        <attribute name="title" type="string" use="required"/>
```