XQuery, Part II

Quantified Expressions

XQuery provides full first-order logic support for building expressions. In particular, XQuery contains expressions for *existentially quantified* and *universally quantified* boolean statements.

Existential quantification. The format of the existentially quantified expression is:

```
some <Var> in <Expression1> satisfies <Expression2>
```

This expression is evaluated as follows:

- `<Expression1>` is evaluated.
- for each value of variable `<Var>` taken from the result of evaluating `<Expression1>`, `<Expression2>` is evaluated.
- If `<Expression2>` is satisfied on least for one value of `<Var>`, `true` is returned. Otherwise, `false` is returned.

Universal quantification. The format of the universaly quantified expression is:

```
every <Var> in <Expression1> satisfies <Expression2>
```

This expression is evaluated as follows:

- `<Expression1>` is evaluated.
- for each value of variable `<Var>` taken from the result of evaluating `<Expression1>`, `<Expression2>` is evaluated.
- If `<Expression2>` is satisfied on all values of `<Var>`, `true` is returned. Otherwise, `false` is returned.

Examples. Consider the following XML document "cafes.xml":

```
<root>
<cafe name="SubJoint">
   <item><n>salad</n><p>3.95</p></item>
   <item><n>hot dog</n><p>1.50</p></item>
</cafe>
<cafe name="Pizza haven">
   <item><n>salad</n><p>3.75</p></item>
   <item><n>garlic bread</n><p>2.50</p></item>
</cafe>
</root>
```

The following expression

```
some $c in doc("cafes")//cafe
satisfies  $c/item/p > 2
```

returns `true` on the `"cafes.xml"` input. Indeed, the XML file does describe both cafes as having at least one item which is priced about $2.

At the same time, its universal equivalent:

```
every $c in doc("cafes")//cafe
satisfies  $c/item/p > 2
```

returns `false`. because not all dishes in the cafes are priced above $2.