Lógica Modal - XPath

Isaque Macalam Saab Lima

September 29, 2015

Outline

XPath

O conteúdo destes slides foi baseado no artigo Logical Foundations of XML and XQuery do Maarten Marx.

O que é XPath?

XPath é o fragmento comum da linguagem de consulta (XQuery) e da liguagem de transformação (XSLT) de documentos XML. É a linguagem que expressa os caminhos para navegar dentro do XML.

XML

XML é o formato padrão para representar e trocar dados semi-estruturados na internet.

Dados estruturados vs Dados semi-estruturados

Dados estruturados podem ser representados em tabelas. Dados semi-estruturados que não são estruturados o suficiente para serem representados em uma tabela.

Modos de representar dados não estruturados: XML (XPath) e JSON (JSON-Path).

Expressões XPath

A maioria dos navegadores web (ex: chrome, firefox e ie) conseguem interpretar expressões XPath.

Exemplos:

```
$x("//div//a[contains(@href,'/')]")
$x("//a[contains(@href,'/')]")
$x("//a[contains(@href,'')]")
```

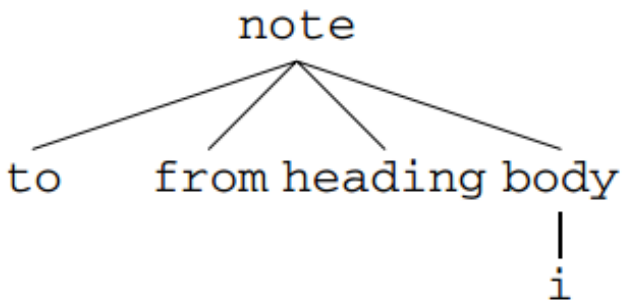
Core XPath

Atualmente XPath pode ser considerada a lógica modal mais conhecida e aplicada, ultrapassando lógicas como CTL e LTL.

Nesse artigo considera-se apenas o "esqueleto" do XML, ignorando o texto entre as tags. Isso permite tratarmos os documentos XML como árvores irmãs ordenadas com nós rotulados. Abordando apenas o core de navegação da linguagem XPath.

Exemplo do esqueleto de um documento XML:

```
<note date='10-Nov-2006'>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget about <i>me</i>
        this weekend!</body>
</note>
```



Core XPath vs PDL

Core XPath e PDL são semelhantes, tendo as seguintes diferenças:

- Enquanto o PDL, em geral, é interpretado sobre um multi-grafo direcionado arbitrário, o core do XPath é um conjunto finito de árvores irmãs ordenadas (documento XML). Existem quatro programas atômicos, correspondentes aos quatro movimentos básicos em uma árvore: child, parent , previous-sibling e next-sibling. Os rótulos dos nós são as tags XML.
- O uso do operado * é restrito a programas atômicos.
- Pequenas diferenças em relação a notações e terminologias.

Sintaxe Core XPath 1.0

A notação da sintaxe foi adaptada para enfatizar a conexão com PDL e evidenciar a existência de uma tradução entre os dois.

Path expressions (definem relações binárias sobre a árvore):

$$\begin{aligned} \alpha ::= \quad & \text{child} \mid \text{parent} \mid \text{previous-sibling} \mid \text{next-sibling} \mid \\ & \text{child*} \mid \text{parent*} \mid \text{previous-sibling*} \mid \text{next-sibling*} \mid \\ & \text{self} \mid \alpha/\beta \mid \alpha U \beta \mid \alpha[\phi] \end{aligned}$$

onde: $\alpha/\beta =$ composição , $\alpha U \beta =$ união e $\alpha[\phi] =$ filtro (α contêm todos os pares (x,y) onde y satisfaz ϕ, ou seja, depois de α, ϕ é verdade).

Node expressions (define o conjunto de nós):

$$\phi ::= \quad p| \ \neg\phi \mid \phi \wedge \varphi \mid <\alpha> \mid$$

onde: $<\alpha>$ = nó em avaliação está no domínio da relação α.

XPath não suporta fórmulas do tipo $<\alpha>\phi$, como PDL, porém essas fórmulas podem ser expressas por: $<\alpha[\phi]>$

Exemplos:

child[<child>]	-	vá para um nó filho que não é folha.
child[note $\wedge$ $\neg$<child[body]>]	-	vá para um nó filho com a tag note que não tenha nenhum filho com a tag body.

Complexidade

A relação entre Core XPath 1.0 e PDL foi utilizada para provar que complexidade de se avaliar uma consulta em XPath é linear.

Na prática uma expressão de caminho α é utilizada da seguinte maneira:

- Avaliamos a expressão em um nó X e retornamos um conjunto de nós que são alcançados por α a partir de X.

Extensões do Core XPath 1.0

Após a utilização do Core XPath 1.0 ficou evidente que algumas aplicações necessitavam de um poder de expressão maior. Iremos abordar duas extensões do XPath que tomam direções opostas.

Regular XPath

A ideia principal é retirar as restrições do operador *.

Problemas:

- Podem existir expressões da Regular XPath que não tem equivalência na lógica de primeira ordem(FO_{tree}).
- Embora seja mais expressiva que FO_{tree} existem formulas FO_{tree^-} (com apenas duas variáveis livres) onde a menor formula equivalente em Regular XPath é muito grande.

Core XPath 2.0

Estende o Core XPath 1.0 nos seguintes operadores:

- Operador de interseção de expressões caminhos, permitindo escrever expressões do tipo: $\alpha \cap \beta$ - "selecione todos os nós que conseguem ser alcançados, a partir do nó atual, por α e β."
- Operador de complementação de expressões de caminho. α - β - "selecione todos os nós que podem ser alcançados, a partir do nó atual, por α e não por β."

- Variáveis e Quantificadores.

```
for $i in Path1 return Path2
```

"atribua para a variável $i algum nó alcançável por Path1, a partir do nó atual, e execute Path2 na variável $i"

Com esses operadores adicionais podemos expressar consultar com $Until$.

O Core XPath 2.0 tem o mesmo poder de expressão que FO_{tree}.

Não existe diferença entre as duas linguagens e existe uma tradução linear entre elas. Isso pode parecer um bom resultado mas também tem o seu lado negativo: a complexidade de FO_{tree}.

A complexidade de se avaliar uma consulta em Core XPath 2.0 é PSpace-complete.

Fim