

# Two semantics for XPath

Philip Wadler, [wadler@research.bell-labs.com](mailto:wadler@research.bell-labs.com)

26 July 1999, revised 4 January 2000

This note presents two semantics for XPath, devised by Phil Wadler after conversations with James Clark at the 19–21 July XSL Face-to-Face meeting. The semantics are somewhat simplified, focussing on the role of axes in selection and on the special functions `position()` and `last()`. Some cases are omitted, and we ignore the coercions between nodesets, numbers, strings, and booleans. The first semantics is given in Figure 1, the second in Figure 2, and both semantics use some auxiliary functions on axes, given in Figure 3. (I consider the second semantics superior, but I gather that James disagrees, so I have presented both.) For a tutorial introduction to denotational semantics and its application to XSL, please see [1].

The first semantics has the advantage of being uniform, in that the three main semantic functions all have the same additional parameters: an axis  $a$ , a context node  $x$ , and a context node set  $S$ ; furthermore, it is always the case that  $x \in S$ . We write  $\mathcal{S}^a[[p]](x, S)$  for the set of nodes selected by pattern  $p$ , and  $\mathcal{Q}^a[[q]](x, S)$  for boolean value of the qualifier  $q$ , and  $\mathcal{E}^a[[e]](x, S)$  for the numerical value of the expression  $e$ . However, the uniformity may obscure more than it reveals: note that the  $S$  parameter is always ignored by  $\mathcal{S}$ , as the line for  $p[q]$  always computes a fresh context node set  $S_1$  to pass in to the semantics of  $q$ . Also, the line for  $p_1/p_2$  is complicated by the need to introduce  $S_1$  in order to satisfy the invariant that the context node is always in the context node set. In any event, using a context node set is overkill, in that the only values that are relevant are the position of the context node in the context node set computed by the call to `position()`, and the size of the context node set computed by the call to `last()`.

The second semantics is simplified in three ways. First, since the context node set is ignored, it is dropped as a parameter to  $\mathcal{S}$ . Second, the context node set is replaced as a parameter to  $\mathcal{Q}$  and  $\mathcal{E}$  by the position of the context node (call this  $k$ ) and size of the context node set (call this  $n$ ). Third, since the position is computed at the first call to  $\mathcal{Q}$ , it is no longer necessary for  $\mathcal{Q}$  or  $\mathcal{E}$  to take an axis parameter. Hence we write  $\mathcal{S}^a[[p]]x$  for the set of nodes selected by pattern  $p$ , and  $\mathcal{Q}[[q]](x, k, n)$  for boolean value of the qualifier  $q$ , and  $\mathcal{E}[[e]](x, k, n)$  for the numerical value of the expression  $e$ .

There are three auxiliary semantic functions on axes. We write  $\mathcal{A}[[a]]x$  for the set of nodes generated by axis  $a$  from context node  $x$ , and  $\mathcal{P}[[a]]$  for the principal node type associated with axis  $a$ , and  $\mathcal{D}[[a]]$  for the direction (forward or reverse node order) associated with axis  $a$ . The last two of these can be computed at compile time, as they are independent of the context node.

Comments on either this note or [1] would be much appreciated.

**Acknowledgements** Thanks to Michael Dyck and Matías Giovannini for spotting, and to Joe English for spotting a significant error.

## References

- [1] P. Wadler, A formal semantics of patterns in XSL, submitted to *Markup Technologies 99*.  
<http://www.cs.bell-labs.com/~wadler/topics/xml#xsl-semantics>

---

$\mathcal{S}$	$: Axis \rightarrow Pattern \rightarrow (Node, Set(Node)) \rightarrow Set(Node)$
$\mathcal{S}^a[p_1   p_2](x, S)$	$= \mathcal{S}^a[p_1](x, S) \cup \mathcal{S}^a[p_2](x, S)$
$\mathcal{S}^a[/math>\text{/}p](x, S)$	$= \mathcal{S}^a[p](root(x), \{x\})$
$\mathcal{S}^a[p_1/p_2](x, S)$	$= \text{let } S_1 = \mathcal{S}^a[p_1](x, S) \text{ in } \{x_2 \mid x_1 \in S_1, x_2 \in \mathcal{S}^a[p_2](x_1, S_1)\}$
$\mathcal{S}^a[a_1 :: p_1](x, S)$	$= \mathcal{S}^{a_1}[p_1](x, S)$
$\mathcal{S}^a[n](x, S)$	$= \{x_1 \mid x_1 \in \mathcal{A}[a]x, nodetype(x_1) = \mathcal{P}[a], name(x_1) = n\}$
$\mathcal{S}^a[*](x, S)$	$= \{x_1 \mid x_1 \in \mathcal{A}[a]x, nodetype(x_1) = \mathcal{P}[a]\}$
$\mathcal{S}^a[\text{text}()](x, S)$	$= \{x_1 \mid x_1 \in \mathcal{A}[a]x, nodetype(x_1) = \mathbf{Text}\}$
$\mathcal{S}^a[p[q]](x, S)$	$= \text{let } S_1 = \mathcal{S}^a[p](x, S) \text{ in } \{x_1 \mid x_1 \in S_1, \mathcal{Q}^a[q](x_1, S_1)\}$
$\mathcal{Q}$	$: Axis \rightarrow Qualifier \rightarrow (Node, Set(Node)) \rightarrow Boolean$
$\mathcal{Q}^a[q_1 \text{ and } q_2](x, S)$	$= \mathcal{Q}^a[q_1](x, S) \wedge \mathcal{Q}^a[q_2](x, S)$
$\mathcal{Q}^a[q_1 \text{ or } q_2](x, S)$	$= \mathcal{Q}^a[q_1](x, S) \vee \mathcal{Q}^a[q_2](x, S)$
$\mathcal{Q}^a[\text{not}(q)](x, S)$	$= \neg \mathcal{Q}^a[q](x, S)$
$\mathcal{Q}^a[p](x, S)$	$= \mathcal{S}^a[p](x, S) \neq \emptyset$
$\mathcal{Q}^a[e_1=e_2](x, S)$	$= \mathcal{E}^a[e_1](x, S) = \mathcal{E}^a[e_2](x, S)$
$\mathcal{E}$	$: Axis \rightarrow Expr \rightarrow (Node, Set(Node)) \rightarrow Number$
$\mathcal{E}^a[e_1 + e_2](x, S)$	$= \mathcal{E}^a[e_1](x, S) + \mathcal{E}^a[e_2](x, S)$
$\mathcal{E}^a[e_1 * e_2](x, S)$	$= \mathcal{E}^a[e_1](x, S) \times \mathcal{E}^a[e_2](x, S)$
$\mathcal{E}^a[\text{position}()](x, S)$	$= \text{let } j = size(\{x_1 \mid x_1 \in S, x_1 \leq_{\text{doc}} x\}) \text{ in}$ $\quad \text{if } \mathcal{D}[a] = \mathbf{forward} \text{ then } j \text{ else } size(S) + 1 - j$
$\mathcal{E}^a[\text{last}()](x, S)$	$= size(S)$
$\mathcal{E}^a[i](x, S)$	$= i$

---

Figure 1: First semantics of XPath

---

$\mathcal{S}$	:	$Axis \rightarrow Pattern \rightarrow Node \rightarrow Set(Node)$
$\mathcal{S}^a[p_1   p_2]x$	=	$\mathcal{S}^a[p_1]x \cup \mathcal{S}^a[p_2]x$
$\mathcal{S}^a[/math>/p]x$	=	$\mathcal{S}^a[p](root(x))$
$\mathcal{S}^a[p_1/p_2]x$	=	$\{x_2 \mid x_1 \in \mathcal{S}^a[p_1]x, x_2 \in \mathcal{S}^a[p_2]x_1\}$
$\mathcal{S}^a[a_1 : : p_1]x$	=	$\mathcal{S}^{a_1}[p_1]x$
$\mathcal{S}^a[n]x$	=	$\{x_1 \mid x_1 \in \mathcal{A}[a]x, nodetype(x_1) = \mathcal{P}[a], name(x_1) = n\}$
$\mathcal{S}^a[*]x$	=	$\{x_1 \mid x_1 \in \mathcal{A}[a]x, nodetype(x_1) = \mathcal{P}[a]\}$
$\mathcal{S}^a[\text{text}()]x$	=	$\{x_1 \mid x_1 \in \mathcal{A}[a]x, nodetype(x_1) = \text{Text}\}$
$\mathcal{S}^a[p[q]]x$	=	$\text{let } S_1 = \mathcal{S}^a[p]x \text{ in}$ $\text{let } n = size(S_1) \text{ in}$ $\{x_1 \mid$ $\quad x_1 \in S_1$ $\quad \text{let } j = size(\{x_2 \mid x_2 \in S_1, x_2 \leq_{\text{doc}} x_1\}) \text{ in}$ $\quad \text{let } k = (\text{if } \mathcal{D}[a] = \text{forward} \text{ then } j \text{ else } n + 1 - j) \text{ in}$ $\quad \mathcal{Q}[q](x_1, k, n)\}$
$\mathcal{Q}$	:	$Qualifier \rightarrow (Node, Number, Number) \rightarrow Boolean$
$\mathcal{Q}[q_1 \text{ and } q_2](x, k, n)$	=	$\mathcal{Q}[q_1](x, k, n) \wedge \mathcal{Q}[q_2](x, k, n)$
$\mathcal{Q}[q_1 \text{ or } q_2](x, k, n)$	=	$\mathcal{Q}[q_1](x, k, n) \vee \mathcal{Q}[q_2](x, k, n)$
$\mathcal{Q}[\text{not}(q)](x, k, n)$	=	$\neg \mathcal{Q}[q](x, k, n)$
$\mathcal{Q}[p](x, k, n)$	=	$\mathcal{S}^{\text{child}}[p]x \neq \emptyset$
$\mathcal{Q}[e_1=e_2](x, k, n)$	=	$\mathcal{E}[e_1](x, k, n) = \mathcal{E}[e_2](x, k, n)$
$\mathcal{E}$	:	$Expr \rightarrow (Node, Number, Number) \rightarrow Number$
$\mathcal{E}[e_1 + e_2](x, k, n)$	=	$\mathcal{E}[e_1](x, k, n) + \mathcal{E}[e_2](x, k, n)$
$\mathcal{E}[e_1 * e_2](x, k, n)$	=	$\mathcal{E}[e_1](x, k, n) \times \mathcal{E}[e_2](x, k, n)$
$\mathcal{E}[\text{position}()](x, k, n)$	=	$k$
$\mathcal{E}[\text{last}()](x, k, n)$	=	$n$
$\mathcal{E}[i](x, k, n)$	=	$i$

Figure 2: Second semantics of XPath

---

---

$\mathcal{A}$	:	$Axis \rightarrow Node \rightarrow Set(Node)$
$\mathcal{A}[\text{child}]x$	=	$children(x)$
$\mathcal{A}[\text{parent}]x$	=	$parent(x)$
$\mathcal{A}[\text{descendant}]x$	=	$children^+(x)$
$\mathcal{A}[\text{ancestor}]x$	=	$parent^+(x)$
$\mathcal{A}[\text{self}]x$	=	$\{x\}$
$\mathcal{A}[\text{attribute}]x$	=	$attributes(x)$
$\mathcal{A}[\text{namespace}]x$	=	$namespaces(x)$
$\mathcal{P}$	:	$Axis \rightarrow Nodetype$
$\mathcal{P}[\text{child}]$	=	<b>Element</b>
$\mathcal{P}[\text{parent}]$	=	<b>Element</b>
$\mathcal{P}[\text{descendant}]$	=	<b>Element</b>
$\mathcal{P}[\text{ancestor}]$	=	<b>Element</b>
$\mathcal{P}[\text{self}]$	=	<b>Element</b>
$\mathcal{P}[\text{attribute}]$	=	<b>Attribute</b>
$\mathcal{P}[\text{namespace}]$	=	<b>Namespace</b>
$\mathcal{D}$	:	$Axis \rightarrow Direction$
$\mathcal{D}[\text{child}]$	=	<b>forward</b>
$\mathcal{D}[\text{parent}]$	=	<b>reverse</b>
$\mathcal{D}[\text{descendant}]$	=	<b>forward</b>
$\mathcal{D}[\text{ancestor}]$	=	<b>reverse</b>
$\mathcal{D}[\text{self}]$	=	<b>forward</b>
$\mathcal{D}[\text{attribute}]$	=	<b>forward</b>
$\mathcal{D}[\text{namespace}]$	=	<b>forward</b>

Figure 3: Auxiliary functions

---