Evaluating XPath axes

In a naive, straightforward manner, XPath axes can be evaluated in using the iterations of four DOM API methods: `getFirstChild()`, `getParent()`, `getNextSibling()` and `getPreviousSibling()`.

When considering XPath, the term *context node* refers to the node currently being observed by the algorithm. All absolute XPath expressions start with a single context node - the root. The effect of each axis can explained by describing the function convertin the context node into the answer node set.

We use OO notation to explain the semantics of the XPath axes below. x is am XML `Node`, the context node for the axis operation. If a method call is followed by * superscript, it means the call can be repeated 0 or more times.

$$self(x) ::= x$$

$$child(x) ::= x.getFirstChild().getNextSibling()*$$

$$parent(x) ::= x.getParent()$$

$$ancestor(x) ::= x.getParent().getParent()^*$$

$$ancestor-or-self(x) ::= x.getParent()^*$$

$$descendant(x) ::= x.getFirstChild().(getFirstChild()^*.getNextSibling()^*)^*$$

$$descendant{-}or{-}self(x) ::= x \cup x.getFirstChild().(getFirstChild()^*.getNextSibling()^*)^*$$

$$following-sibling(x) ::= x.getNextSibling().getNextSibling()^*$$

$$preceding-sibling() ::= x.getPreviousSibling().getPreviousSibling()^*$$

$$following(x) ::= x.getParent()^{*}.getNextSibling().getNextSibling()^{*}.descendant{-}or{-}self(x)$$

$$preceding(x) ::= x.getParent()^{*}.getPreviousSibling().getPreviousSibling()^{*}.descendant{-}or{-}self(x)$$

Using *getChildren()* some of these axis descriptions can be simplified even further.

$$child(x) ::= x.getChildren()$$
$$descendant(x) ::= x.getChildren().getChildren()^{*}$$