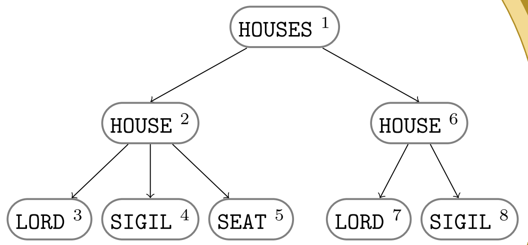
Indexing XML Documents

*An **XML** document represents a **tree** hierarchical structure. Methods of indexing tree data structures have not been researched in so many details as in the case of methods for indexing **texts**. We show that indexes based on **automata** can be used **effectively** for the purpose of **indexing** XML documents.*

```
<HOUSES>
 <HOUSE>
  <LORD>...</LORD>
  <SIGIL>...</SIGIL>
  <SEAT>...</SEAT>
 </HOUSE>
 <HOUSE>
  <LORD>...</LORD>
  <SIGIL>...</SIGIL>
 </HOUSE>
</HOUSES>
```



Problem Statement

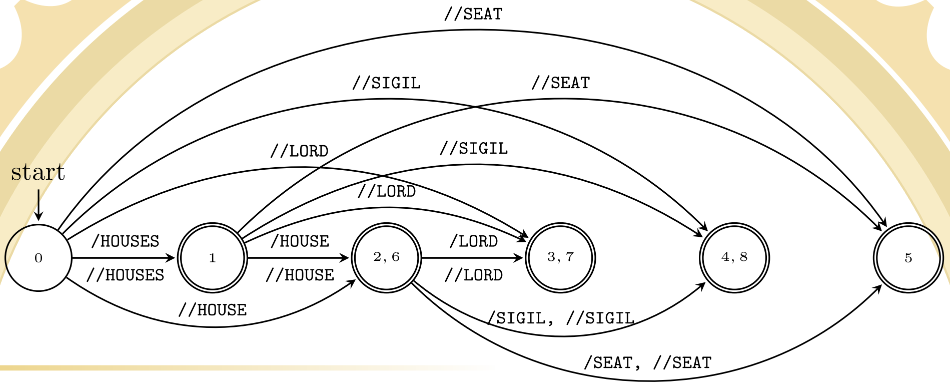
Indexing a data subject preprocesses the subject and constructs an index that allows to efficiently answer queries related to the content of the subject. Therefore, indexing the structure of XML data is an effective way to accelerate the XML query languages processing.

Automata Approach

We introduce new methods for the purpose of indexing XML documents based on the standard theory of formal languages and automata. It makes it well understandable and convenient for combinations to construct indexes for unions, intersections and other operations.

The following finite automata have been proposed:

- **Tree String Paths Automaton (TSPA)** speeds up the evaluation of linear XPath queries using child axis (/) only.
- **Tree String Path Subsequences Automaton (TSPSA)** efficiently evaluates exponential number of linear XPath queries where just descendant-or-self axis (//) is used.
- **Tree Paths Automaton (TPA)** is designed to process a significant fragment of XPath queries, which may use any combination of / and // axes.



Results

	TSPA	TSPSA	TPA
# queries supported	linear	exponential	exponential
searching phase	O(m)	O(m)	O(m)
# states in DFA	O(n)	$O(h^k)$ $^*O(h.2^k)$	experimental evaluation

n ... number of nodes in XML tree model
m ... length of a query
h ... height of XML tree model
k ... number of leaves in XML tree model

* for common XML documents

Eliška Šestáková
supervised by Jan Janoušek