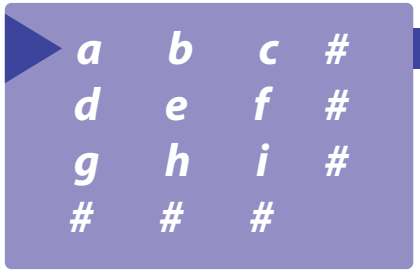
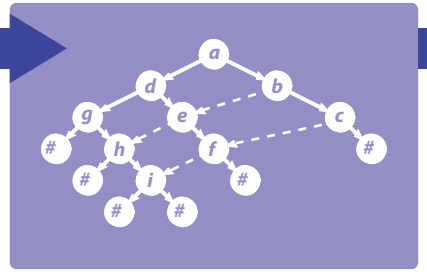
# Implementation of 2D Pattern Matching Algorithm using Pushdown Automata

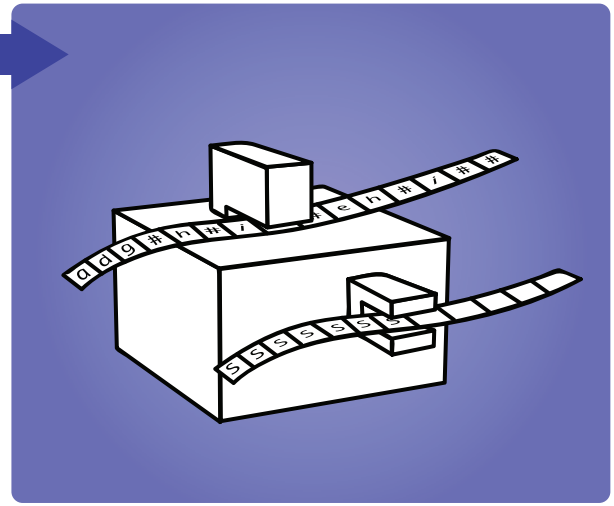



A searched pattern is first extended with special # symbols



The extended pattern is transformed into a tree


The tree is linearised into a string using prefix notation



The string is processed by a push-down automaton constructed from a reference picture.

If the automaton accepts the input, the pattern is matched.

## Our goals

- analysis and optimisation of the original algorithm
- implementation of the algorithm using platform-independent tools
- to benchmark the algorithm using various reference pictures and patterns

## Main drawbacks of the algorithm

- the amount of data the automaton has to process *(the picture-to-tree transformation creates a lot of redundant data)*
- non-determinism of the indexing automaton

## Optimisation

- the redundancies were removed for faster processing
- the number of non-deterministic transitions was reduced to only one for simpler simulation
- two non-deterministic automaton simulation approaches using sets of deterministic sub-automata were proposed *(BFS, DFS)*

## Implementation

- ANSI C compliant
- consists of several modules
  - *resource loader*
  - *on-the-fly input generator*
  - *automaton simulator*
- both approaches of automaton simulation were implemented

## Testing

- the testing was performed on Intel x86 and Sun SPARC platforms
- focused on algorithm performance for various categories of pictures and comparison of automaton simulation approaches
- categories: colour and grayscale photographs, OCR text, random noise...
- DFS approach gave better results in all tests
- time complexity $O(n^2)$, space complexity $O(n)$ *(for the DFS)*

Jiří Šembera
Supervisor: Ing. Jan Žďárek, Ph.D.

Department of Computer Science and Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague