UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

Regular Expression matching with Thompson-algorithm

Renata Hodovan
University of Szeged

Introduction

- Regular expressions
  - Searching patterns in strings
  - Validation
    - E-mail
      - » ^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$
    - Phone-number
      - » \+[0-9]{4}\/[0-9]{7} -> +3630/3422134

The Engine

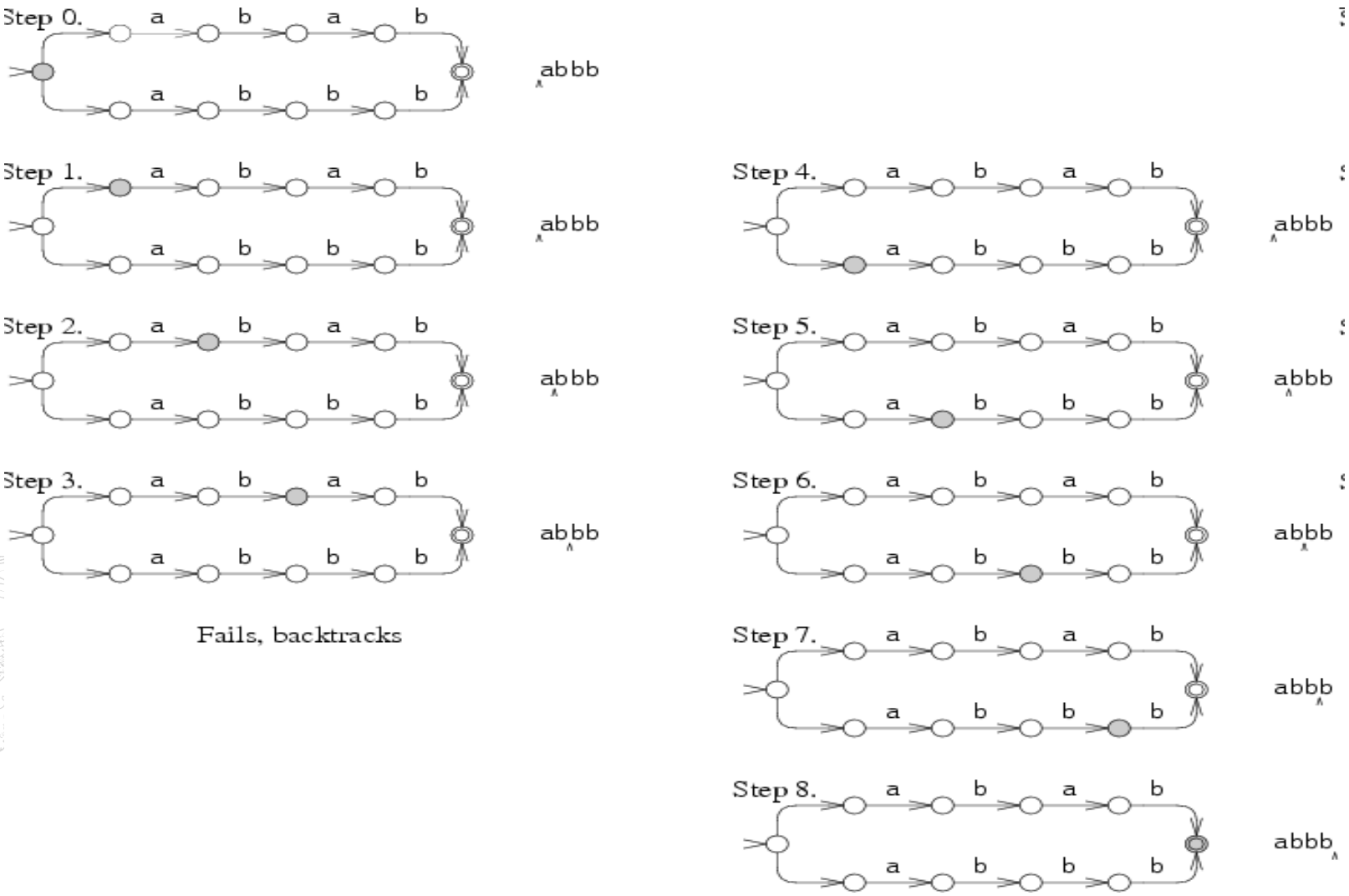
- Regular expression implemetation based on two major families of algorithms:
  - NFA (Non-deterministic Finite Automata)
    - (PERL, Ruby, Phyton, PHP)
  - DFA (Deterministic Finite Automata)
    - (grep, egrep, awk)

NFA

- Non-deterministic Finite Automata
- Recursive algorithm
- Only a single path at a time
- Backtracking
- „Pathological” cases
- PCRE, YARR

$$a\,?^{n}\,a^{n} \rightarrow a^{n}\,b^{n}$$

Example: abbb to /abab|abbb/

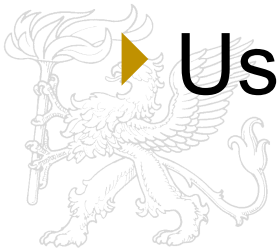


DFA

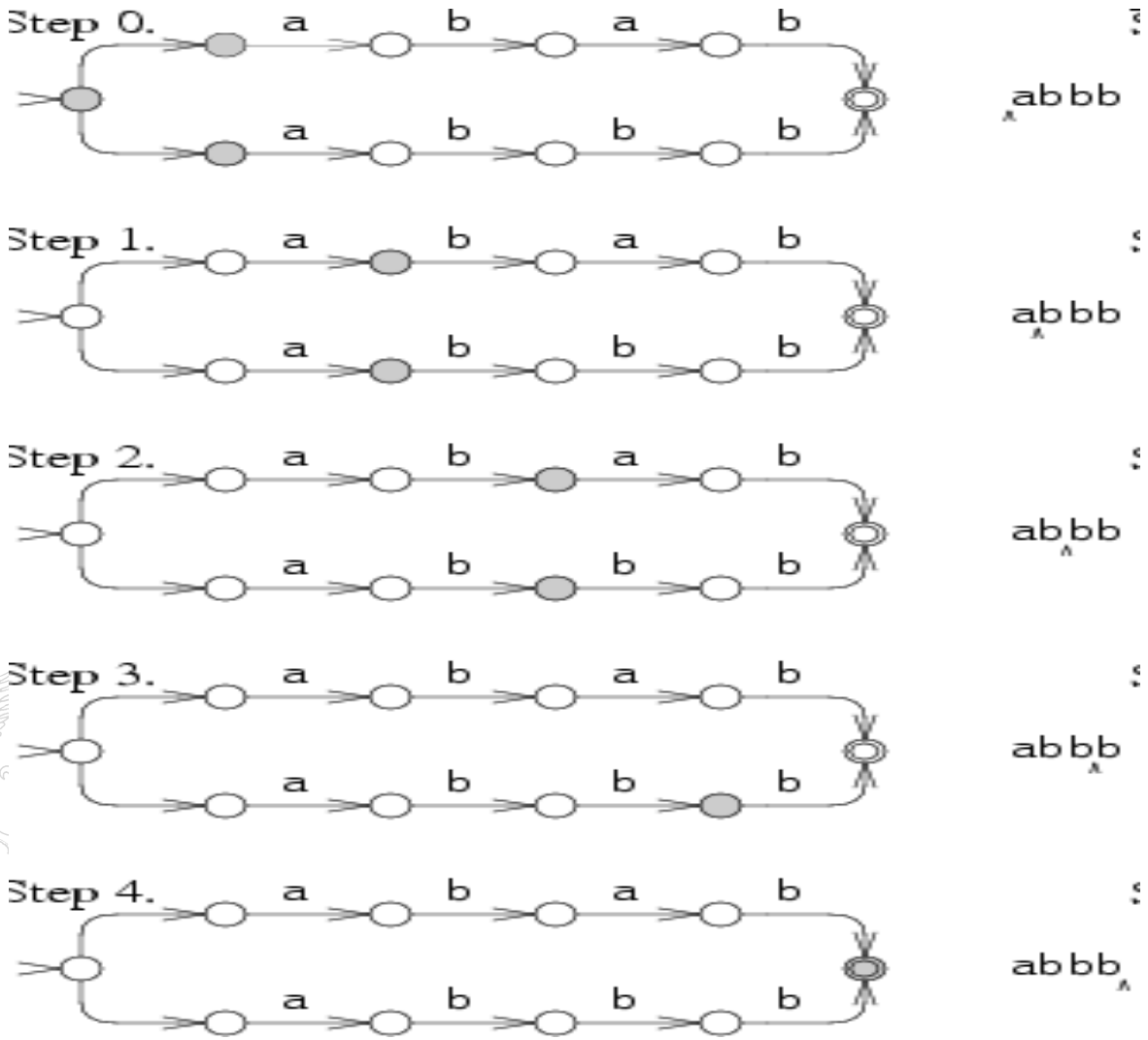
- Deterministic Finite Automata
- Store all possible states at the same time
- No backtrack
- Breadth first search
- Thompson-algorithm

Thompson-algorithm

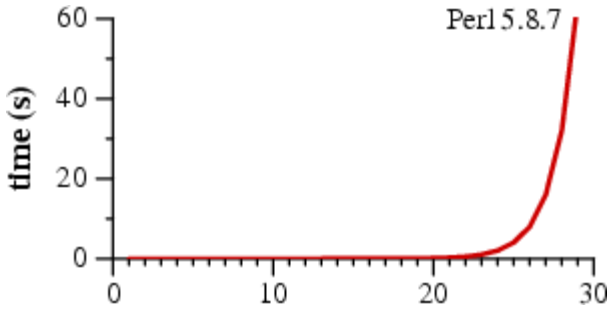
- Introduced by Ken Thompson in 1960
- Used by UNIX system (grep, egrep, awk, etc.)
- Linear running time
- No „pathological” cases
- Using DFA instead of NFA

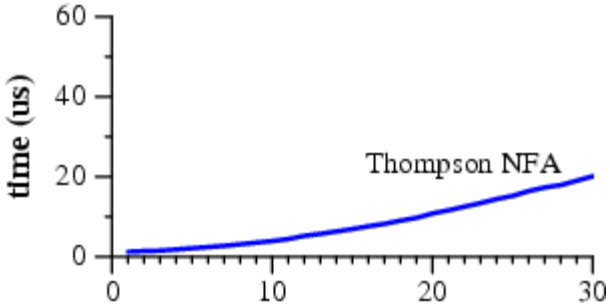


Example: abbb to /abab|abbb/



Thompson vs recursive





- Regex: $a\,?^{n}\,a^{n} \rightarrow a^{n}$
- Input: 29-character string
- Perl (recursive): >60 sec
- Thompson: 20 micro(!)sec

ECMA's restriction

- Pretend recursive algorithm
- Enumerate subpattern
- Backreferences

Finished tasks

- Converting NFA to DFA
- Both Interpreter and JIT
- They work, but haven't finished
- No benchmarking yet

Thanks for your patience!