Automata Theory :: Word Matching

Jörg Endrullis

Vrije Universiteit Amsterdam

Word (String) Matching (Thompson, 1968)

The input:

- a word u
- a regular expression r

Question: Does u contain a subword in $L(r)$?

The following algorithm answers this question.

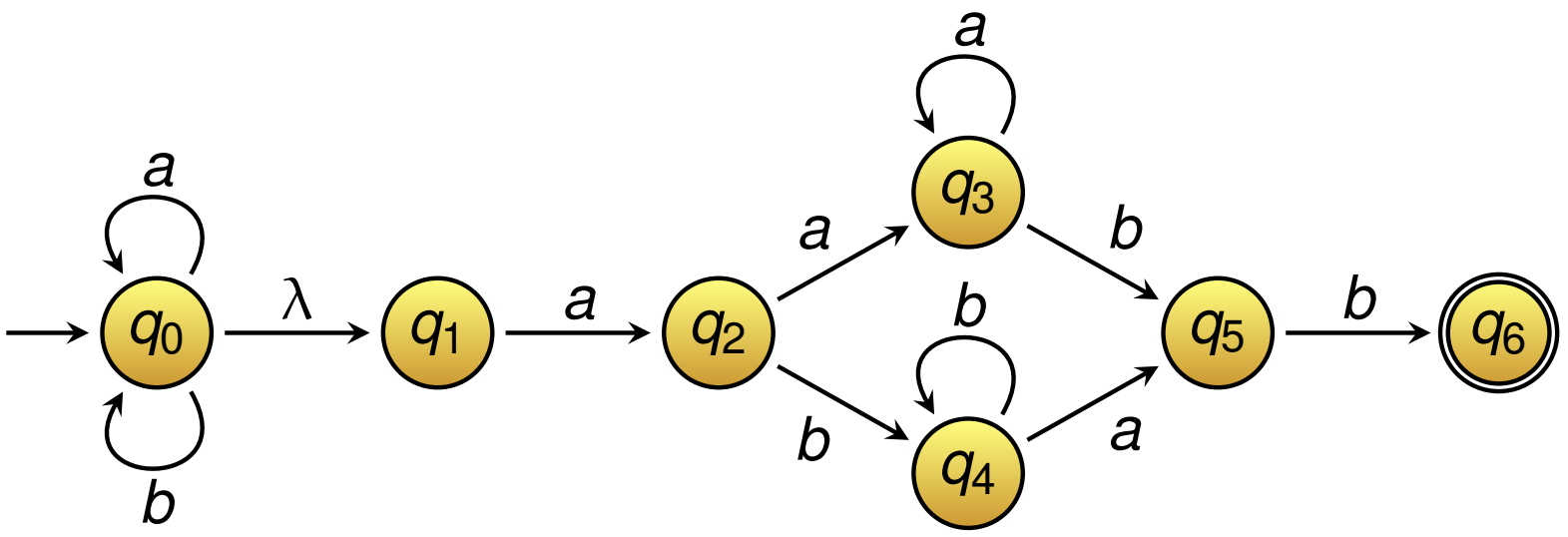
1. Transform the regular expression $\Sigma^* \cdot r$ into an NFA.
2. Compute '**on-the-fly**' path of u in the corresponding DFA.
3. Terminate as soon as a final state is reached.

The algorithm is used for example in grep in Unix.

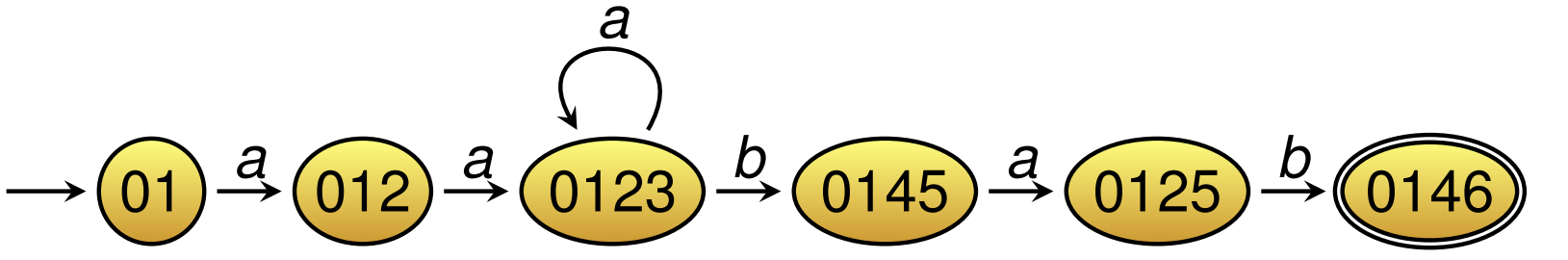
Worst-case time complexity: $O(|r| \cdot |u|)$

Word Matching Example

Regular expression $r = a(aa^*b + bb^*a)b$ gives rise to the NFA:



We match r with $u = aaababbb$.



a	a	a	b	a	b	b	b
q_0	q_0	$q_0\,q_1$	q_2	q_4	q_5	q_6	

Word (String) Matching (Thompson, 1968)

The standard regular expression libraries of

- Java,
- Perl,
- PHP,
- Python

do not use the efficient algorithm from the last slide.

They use a **backtracking** algorithm with worst-case complexity

exponential time (in $|u|$)

Ken Thompson

Matching algorithm has been developed by **Ken Thompson**.

Won the Turing award in 1983 together with **Dennis Ritchie** for the operating system Unix. Dennis Ritchie (1941-2011) has also invented the programming language C.