From Regular Expression Matching to Parsing

Philip Bille
Inge Li Gørtz

Regular Expressions

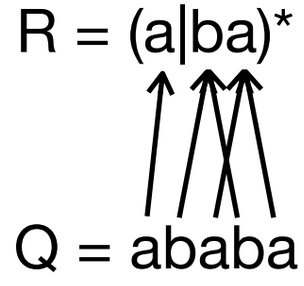
$$R = (a|ba)^*$$

$$L(R) = \{\varepsilon, a, aa, ba, aaa, aba, baa, aaaa, aaba, abaa, baaa, baba, ...\}$$

Regular Expression Matching and Parsing

R = (a|ba)*

Q = ababa



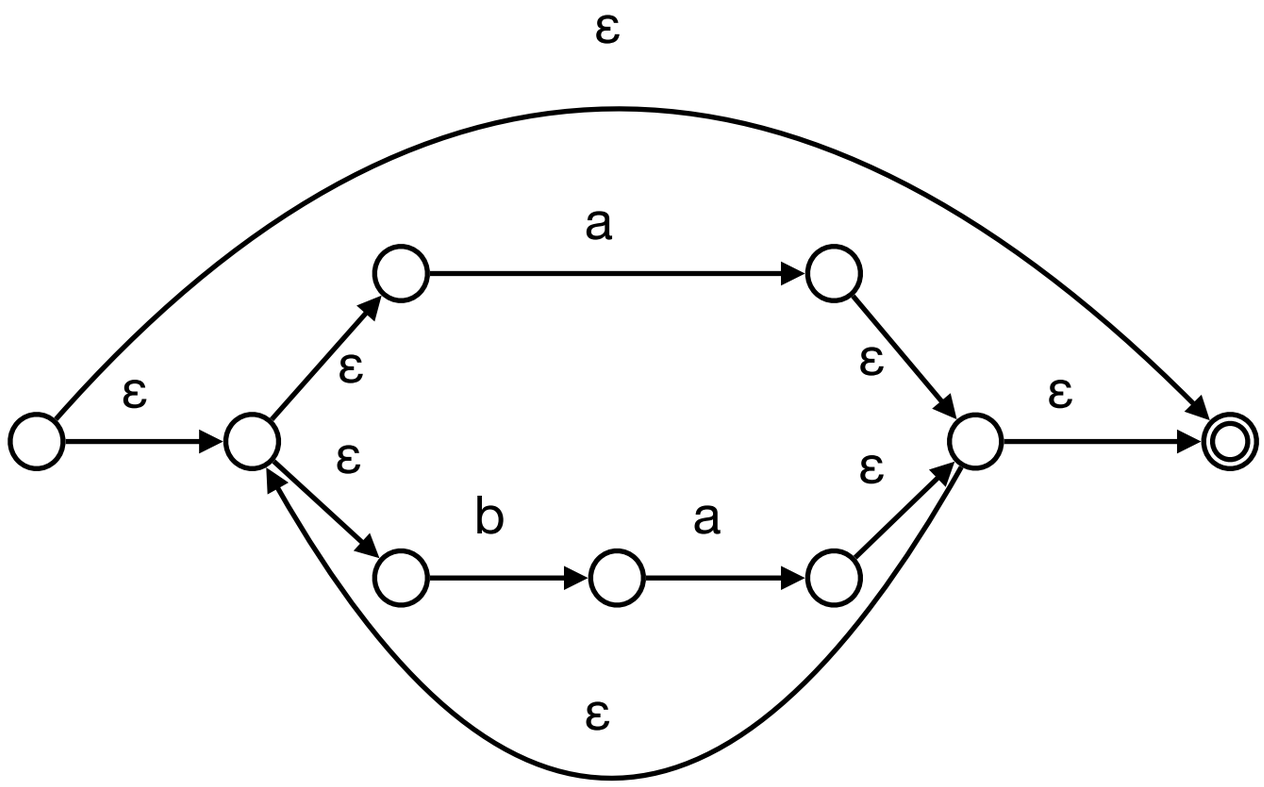
Matching: Is Q in L(R)?

Parsing: *How* is Q in L(R)?

Regular Expression Matching

R = (a|ba)*

Q = ababa

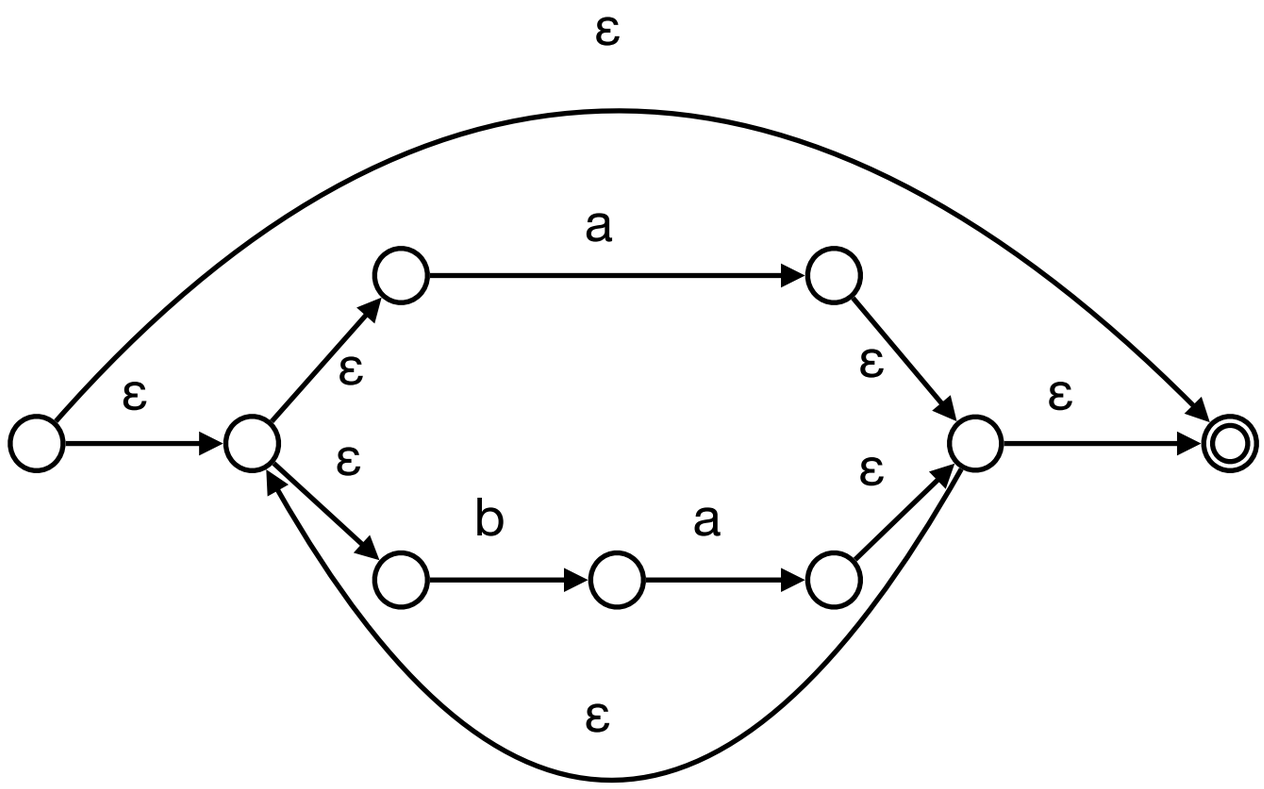


- O(nm) time and O(m) space [Thompson 1968]
- O(nm/polylog n) time improvements [Myers 1992, B. 2006, B.-Farach-Colton 2008, B.-Thorup 2009, 2010]
- $\Omega((nm)^{1-\varepsilon})$ conditional lower bound [Backurs-Indyk 2016]

Regular Expression Parsing

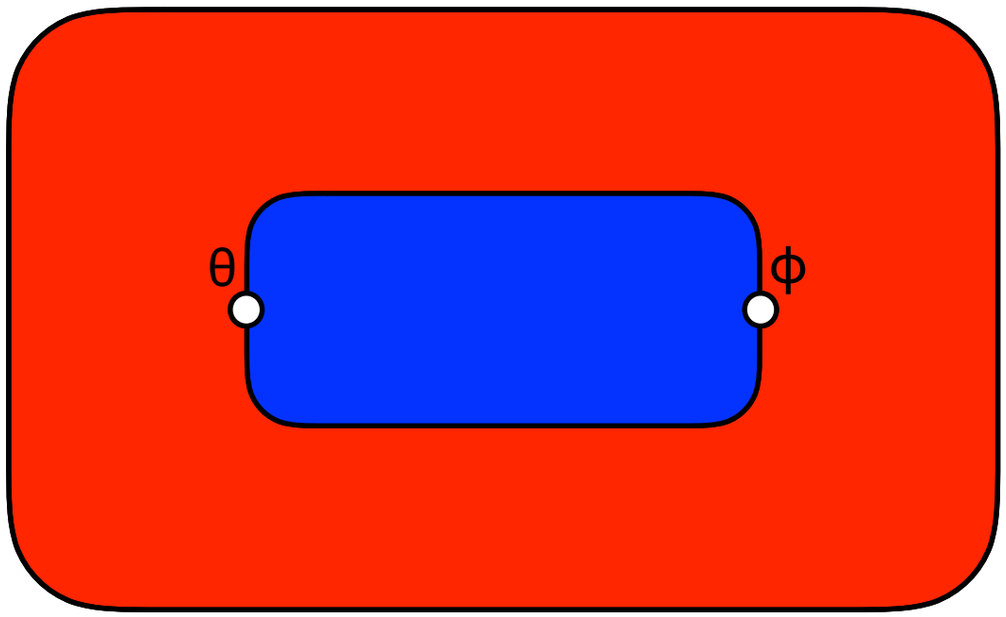
R = (a|ba)*

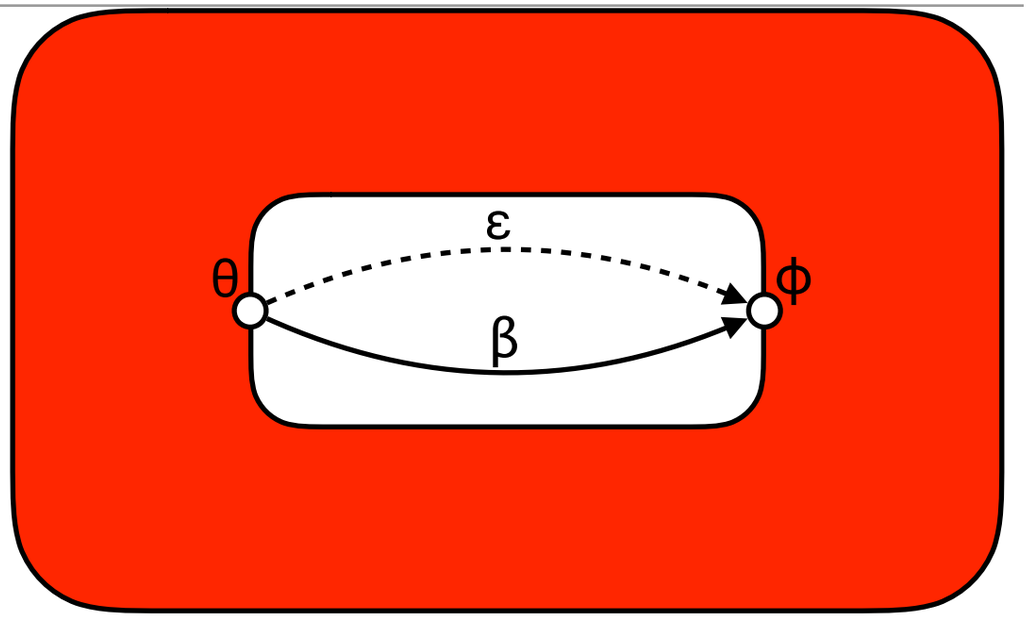
Q = ababa

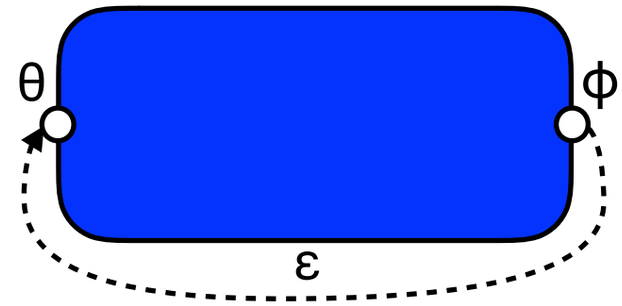


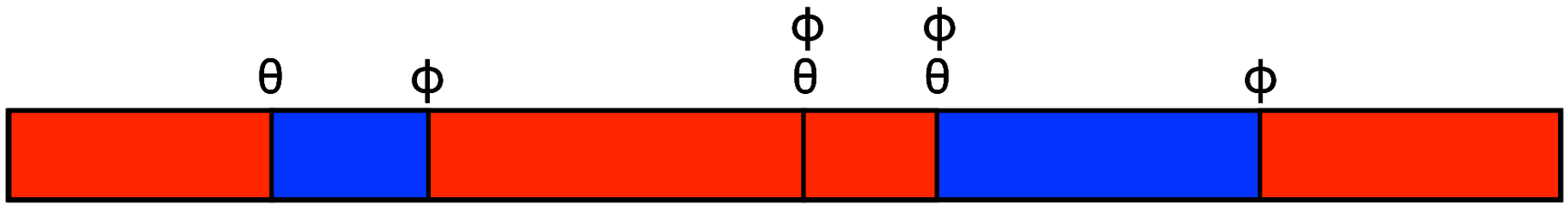
- O(nm) time and O(nm) space (Thompson + backtracking).
- Other approaches [Kearns 1991, Dube-Feeley 2000, Laurikari 2000, Frisch-Cardelli 2004, Nielsen-Henglein 2011, Sulzmann-Lu 2012].
- **This paper:** General technique to convert regular expression matching algorithm to solve regular expression parsing at no asymptotic cost.

Divide



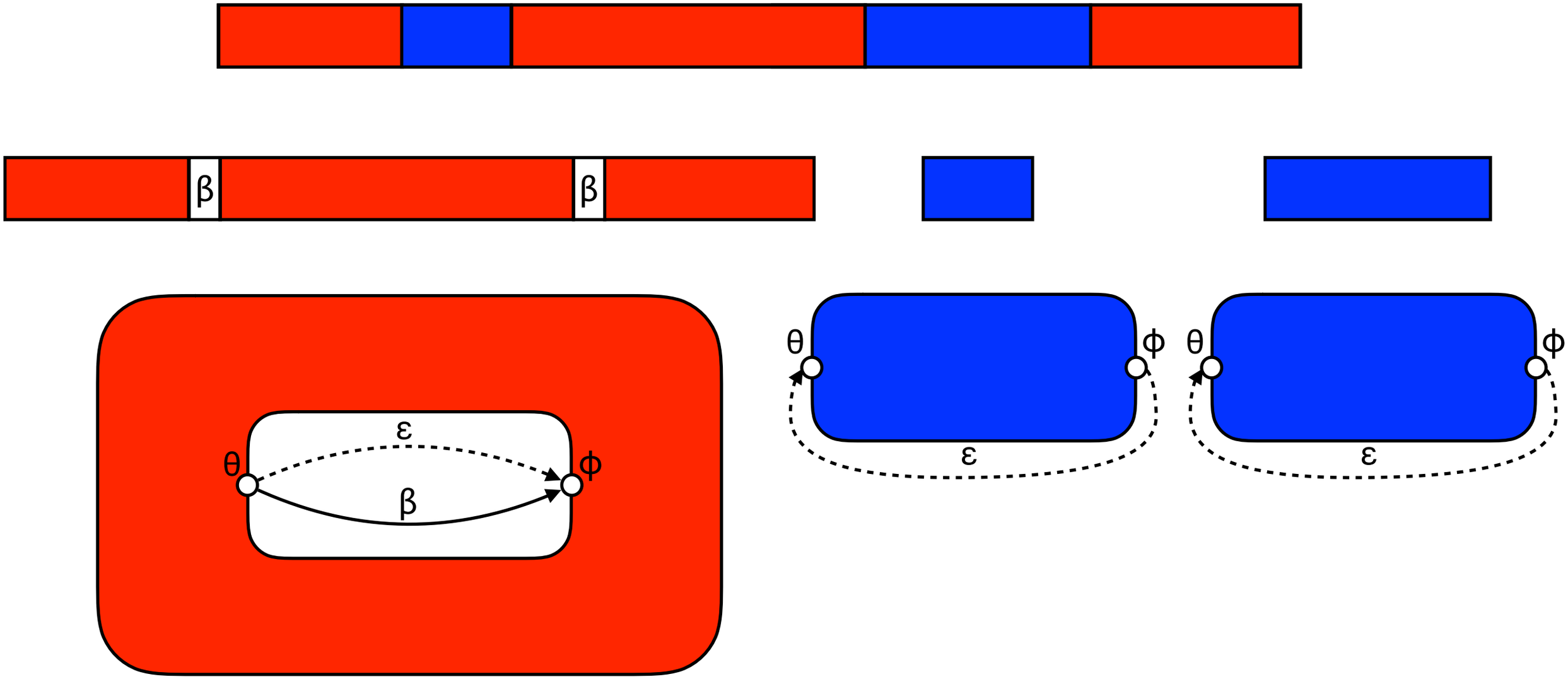






- O(nm) time and O(n+m) space.

Conquer



- O(nm) time and O(n log m) space.
- Compact encoding for mappings + heavy path decomposition ⇒ O(n+m) space.

Results

- General technique to convert regular expression matching algorithms to solve regular expression parsing at no asymptotic cost.
  - For Thompson's algorithm: O(nm) time and O(n + m) space solution for regular expression parsing.
  - For *almost all* existing faster regular expression matching: same time and O(n + m) space for regular expression parsing.