RW178: Implementation and Application of Automata, 2006 Week 7 Lecture 1

L. van Zijl

Department of Computer Science
University of Stellenbosch

2006

Pattern Matching – Linear Dictionary

References:

1. Handbook of Formal Languages, Chapter 9.
2. Handbook of Algorithms, pp 11-10 to 11-13.

Pattern Matching in Dictionaries

- Pattern matching – finding occurrences of a pattern in a given text
- Exact pattern matching – classic solution Aho-Corasick
- Two phases – preprocessing pattern phase, search phase
- Dictionary-matching problem (Aho-Corasick): *Given a finite set X of words called the dictionary, preprocess it in order to locate words of X that occur in any given text y.*
- Unix fgrep

Pattern Matching in Dictionaries

The Aho-Corasick Algorithm

- Dictionary X size m, text y size $n \longrightarrow$ time $O(m+n)$
- Automaton stores prefixes
- At given position in text, current state identified with set of pattern prefixes ending here
- Automaton $D(X)$ with states in 1–1 correspondence with prefixes of X: requires $O(m \times \text{card}\Sigma)$ space
- Aho-Corasick requires $O(m)$ space – failure function

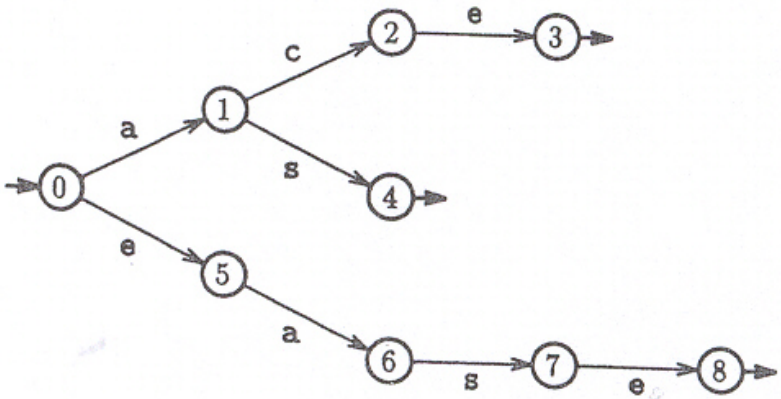
Pattern Matching in Dictionaries

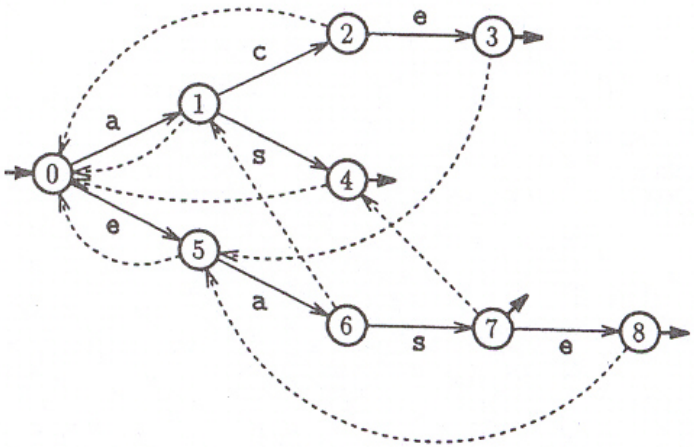
The Aho-Corasick Algorithm

- fail(p) = state identified with longest proper suffix of the prefix identified with p that is also prefix of a string of X
- Step 1: Build trie-like automaton recognizing X
- Step 2: Pre-compute failure function (making more states final).
- Step 3: Traverse $D(X)$

Pattern Matching in Dictionaries

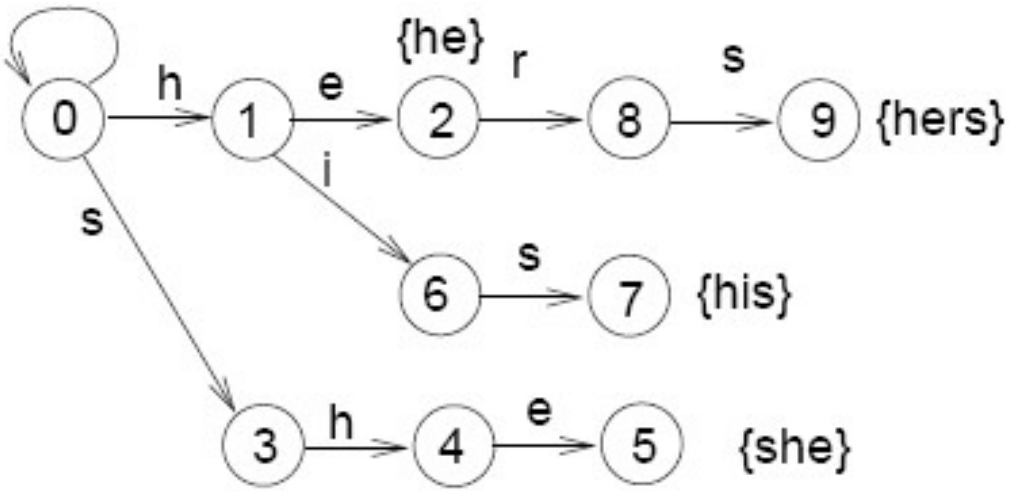
The Aho-Corasick Algorithm: Example

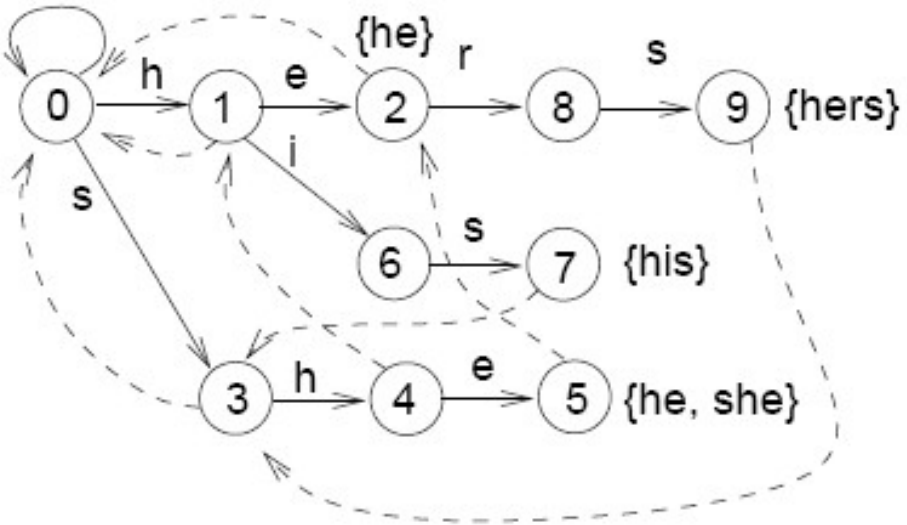




Pattern Matching in Dictionaries

The Aho-Corasick Algorithm: Another Example





Pattern Matching in Dictionaries

Homework

Homework: Implement the Aho-Corasick algorithm.