# CSE 322: Introduction to Formal Models in Computer Science

## Pattern Matching

Paul Beame



## Pattern Matching

- Given
  - a string, **s**, of **n** characters
  - a pattern, **p**, of **m** characters
  - usually **m<<n**
- Find
  - all occurrences of the pattern **p** in the string **s**
- Obvious algorithm:
  - try to see if **p** matches at each of the positions in **s**, stopping at a failed match



String s = x y x x y x y x y y x y x y y y x y x y x x

Pattern p = x y x y y x y x y x



String s = x y x x y x y x y y x y x y y y x y x y x x

x y x y y x y x y x



String s = x y x x y x y x y y x y x y y y x y x y x x

x y x y

x y x y y x y x y x



String s = x y x x y x y x y y x y x y y y x y x y x x

x y x y

x

x y x y y x y x y x

String s = x y x x y x y x y y x y x y x y y x y x x
x y x y
x
x y
x y x y y x y x y x
7

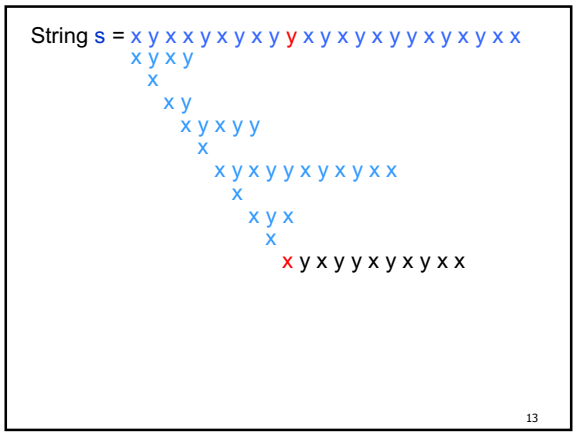
String s = x y x x y x y x y y x y x y x y y x y x x
x y x y
x
x y
x y x y y
x y x y y x x y x y x
8

String s = x y x x y x y x y y x y x y x y y x y x x
x y x y
x
x y
x y x y y
x
x y x y y x y x y x x
9

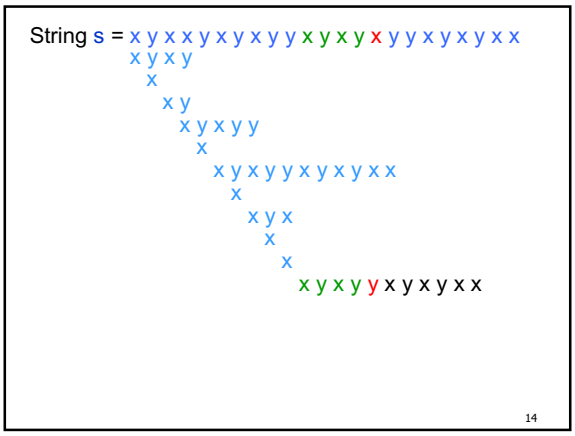
String s = x y x x y x y x y y x y x y x y y x y x x
x y x y
x
x y
x y x y y
x
x y x y y x y x y x x
10

String s = x y x x y x y x y y x y x y x y y x y x x
x y x y
x
x y
x y x y y
x
x y x y y x y x y x x
11

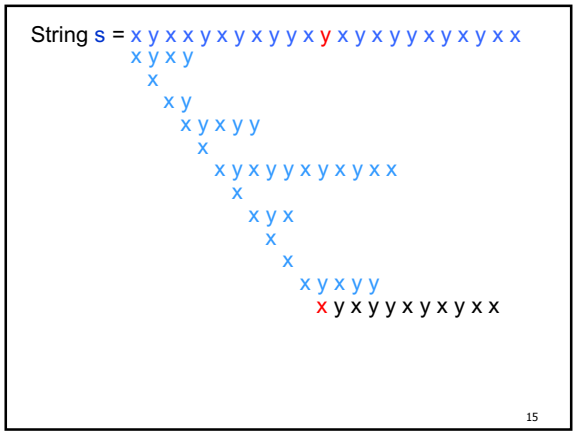
String s = x y x x y x y x y y x y x y x y y x y x x
x y x y
x
x y
x y x y y
x
x y x y y x y x y x x
12



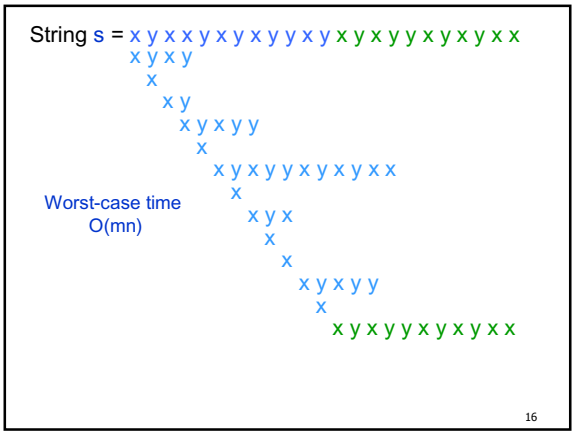




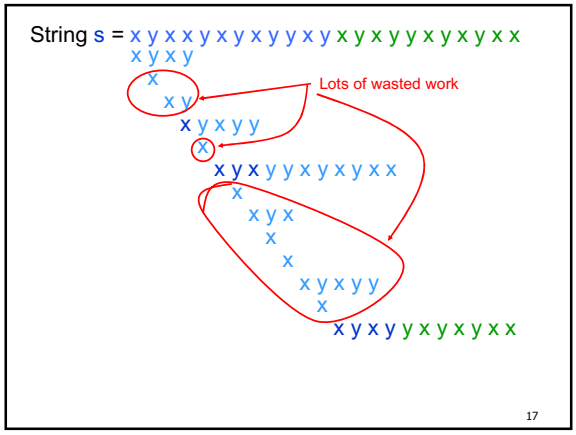












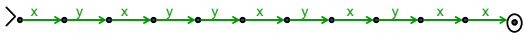


## Better Pattern Matching via Finite Automata

- Build an DFA for the pattern (preprocessing) of size O(m)
  - Keep track of the 'longest match currently active'
  - The DFA will have only O(m) states
- Run the DFA on the string O(n)
- Obvious construction method for DFA will be $O(m^2)$ but can be done in O(m) time.
- Total O(m+n) time

## Building a DFA for the pattern

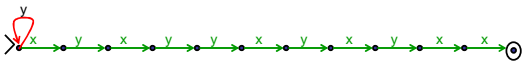
Pattern p=x y x y y x y x y x



19

## Preprocessing the pattern

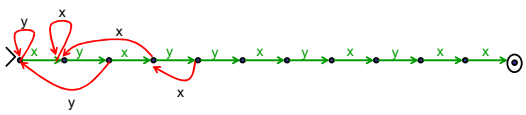
Pattern p=x y x y y x y x y x



20

## Preprocessing the pattern

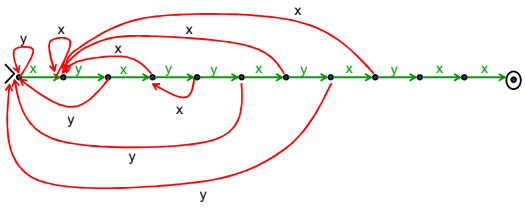
Pattern p=x y x y y x y x y x



21

## Preprocessing the pattern

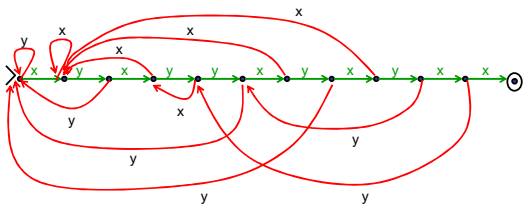
Pattern p=x y x y y x y x y x



22

## Preprocessing the pattern

Pattern p=x y x y y x y x y x



23

## Knuth-Morris-Pratt Algorithm

- Once the preprocessing is done there are only n steps on any string of size n
  - just follow your nose
- Obvious algorithm for doing preprocessing is $O(m^2)$ steps
  - still usually good since m<<n
- Knuth-Morris-Pratt Algorithm can do the pre-processing in O(m) steps
  - Total O(m+n) time

24

## Generalizing

- Can search for arbitrary combinations of patterns not just a single pattern
  - Build NFA for pattern then convert to DFA 'on the fly'. (Compare DFA constructed with subset construction for the obvious NFA.)
- Typical text searches are based on finite automata designs
  - Perl builds this in as a first-class component of the programming language
  - grep

25