Announcements

• Reading assignments
  — 7th Edition, Section 13.4
  — 6th Edition, Section 12.4
  — 5th Edition, Section 11.4
• Next week
  — 7th edition: 2.5 (Cardinality) + p. 201 and 13.5
  — 5th edition: Pages 233-236 (Cardinality) and 11.5
• Topic list and sample final exam problems have been posted
• Comprehensive final, roughly 67% of material post midterm
• Review session, Saturday, December 8, 10 am – noon (tentatively)
• Final exam, Monday, December 10
  — 2:30-4:20 pm or 4:30-6:20 pm, Kane 220.
  — If you have a conflict, contact instructors ASAP

Last lecture highlights

• NFAs from Regular Expressions
  \((01 \cup 1)^*0\)

Converting an NFA to a regular expression

• Consider the DFA for the mod 3 sum
  — Accept strings from \((0,1,2)^*\) where the digits mod 3 sum of the digits is 0

Splicing out a node

• Label edges with regular expressions
  \[\begin{align*}
  t_0 \rightarrow t_1 \rightarrow t_0 &: \ 10^*2 \\
  t_0 \rightarrow t_2 \rightarrow t_0 &: \ 10^*1 \\
  t_0 \rightarrow t_3 \rightarrow t_2 &: \ 20^*2 \\
  t_0 \rightarrow t_4 \rightarrow t_2 &: \ 20^*1
  \end{align*}\]
Finite Automaton without $t_1$

$R_1$: 0 | $10^*2$
$R_2$: 2 | $10^*1$
$R_3$: 1 | $20^*2$
$R_4$: 0 | $20^*1$

$R_5$: $R_1 | R_2 R_4^* R_3$

Final regular expression: ($0 | 10^*2 | (2 | 10^*1)(0 | 20^*1)^*1 | 20^*2)^*$
A=\{0^n1^n : n \geq 0\} cannot be recognized by any DFA
Consider the infinite set of strings
S=\{\lambda, 0, 00, 000, 0000, ...\}
Claim: No two strings in S can end at the same state of any DFA for A, so no such DFA can exist
Proof: Suppose n \neq m and 0^n and 0^m end at the same state p.
Since 0^n1^n is in A, following 1^n after state p must lead to a final state.
But then the DFA would accept 0^m1^n which is a contradiction.

The set B of binary palindromes cannot be recognized by any DFA
Consider the infinite set of strings
S=\{\lambda, 0, 00, 000, 0000, ...\}
Claim: No two strings in S can end at the same state of any DFA for B, so no such DFA can exist
Proof: Suppose n \neq m and 0^n and 0^m end at the same state p.
Since 0^n10^n is in B, following 10^n after state p must lead to a final state.
But then the DFA would accept 0^m10^n which is a contradiction.

The set P of strings of balanced parentheses cannot be recognized by any DFA

The set P of strings \{1^j \mid j = n^2\} cannot be recognized by any DFA
Suppose 1^j and 1^k reach the same state p with j < k
1^j1^{j(k-1)} must reach an accepting state q
1^k1^{k(k-1)} must reach the same accepting state q
Thus, j + k(k-1) = k^2 - k + j must be a perfect square
Is that possible?

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
    - stop at a failed match and try the next position

Pattern p = x x x x x y
String s = x x x x x x x x y y y y y y x x
Pattern $p = xyxyxyxyxyx$
String $s = xyxyxyxyxyxyxyxyx$

String $s = xyxyxyxyxyxyxyxyx$

String $s = xyxyxyxyxyxyxyxyx$

String $s = xyxyxyxyxyxyxyxyx$

String $s = xyxyxyxyxyxyxyxyx$

String $s = xyxyxyxyxyxyxyxyx$
Better Pattern Matching via Finite Automata

• Build a DFA for the pattern (preprocessing) of size $O(m)$
  - Keep track of the 'longest match currently active'
  - The DFA will have only $m+1$ states
• Run the DFA on the string in $n$ steps
• 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 $p = \text{x y y x y y x y x}$

Preprocessing the pattern

Pattern $p = \text{x y y x y y x y x}$
Preprocessing the pattern

Pattern $p = x \ y \ x \ y \ y \ x \ y \ x \ y \ x \ x$

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 above with subset construction for the obvious NFA.