LING/C SC/PSYC 438/538

Lecture 22

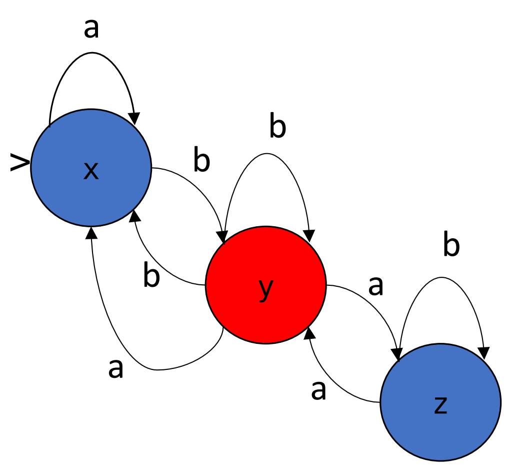
Sandiway Fong

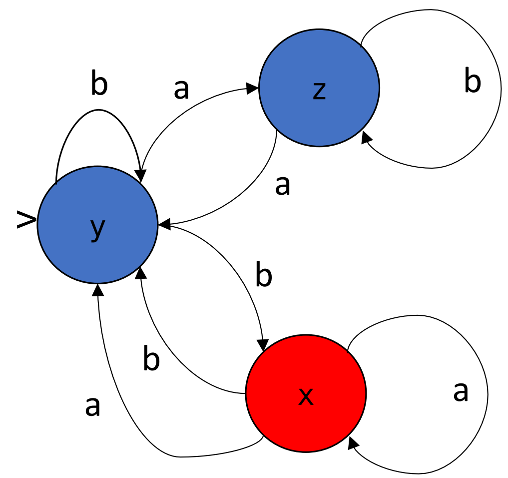
Today's Topics

- Homework 11 Review
- Beyond regular languages:
  1. $\{a^n b^n \mid n \geq 1\}$, and
  2. $\{1^n \mid n \text{ is prime}\}$
- A formal tool: the Pumping Lemma

Homework 11 Review

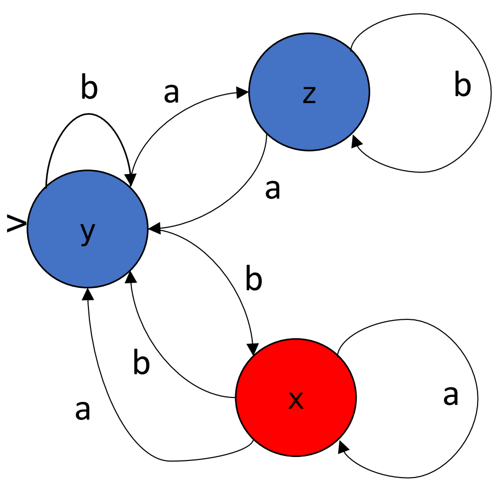
- Q1: $L_R = \{w^R \mid w \in L\}$

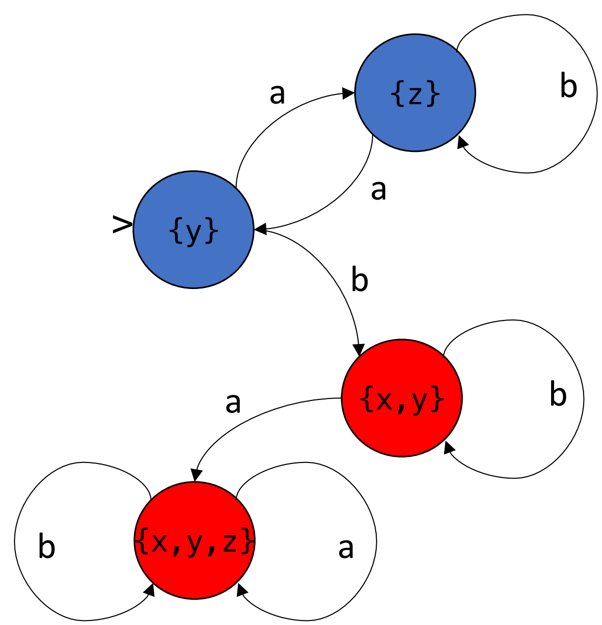




Homework 11 Review

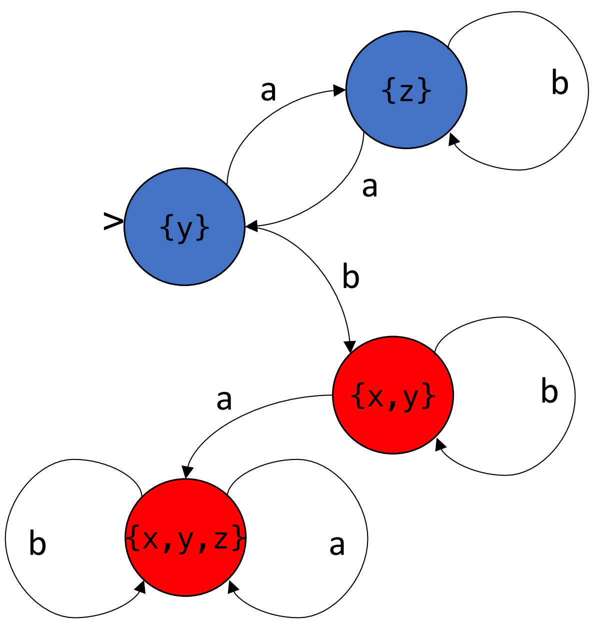
- Q2: convert $L_R = \{w^R \mid w \in L\}$ to a DFSA

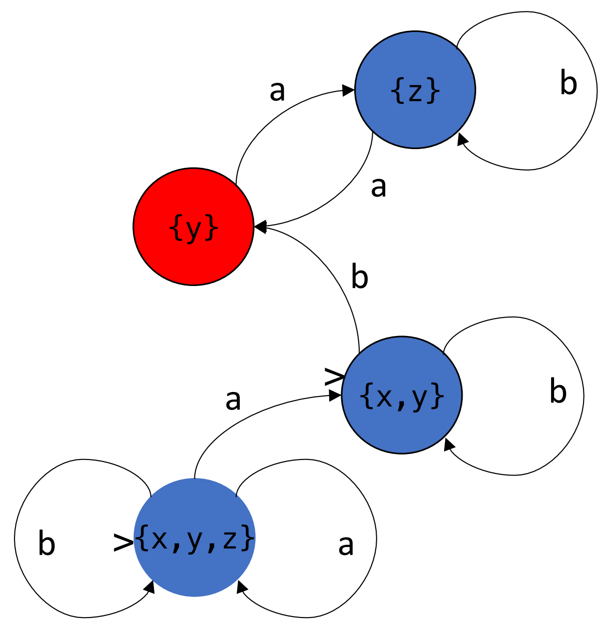




Homework 11 Review

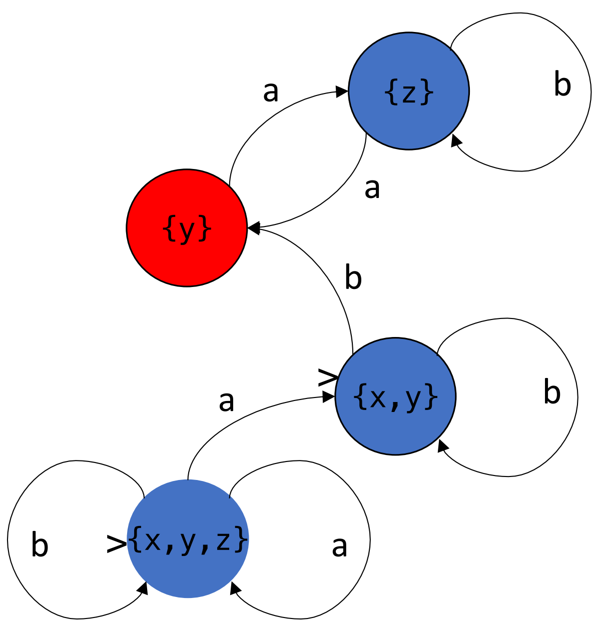
- Q3: $L_{RR} = \{w^R \mid w \in L_R\}$

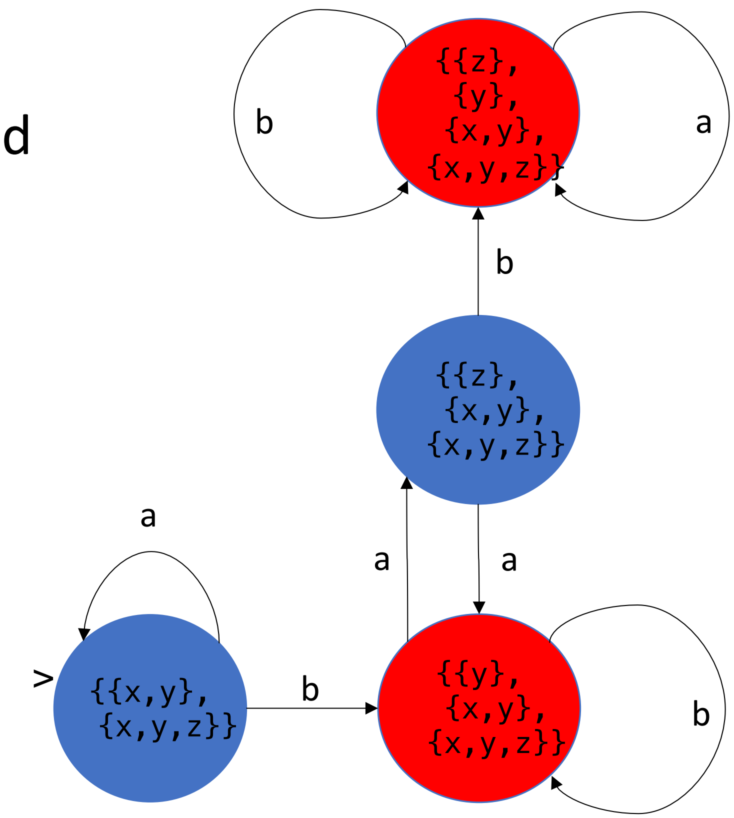




Homework 11 Review

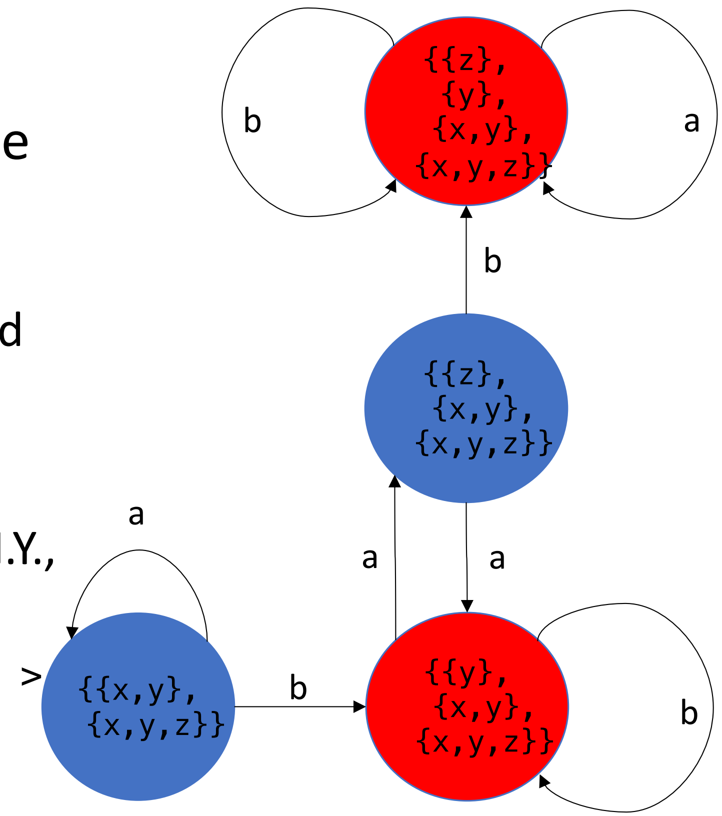
- Q4: $L_{RR} = \{w^R \mid w \in L_R\}$ determinized





State Minimization

- Do you think you could build a machine for L (= L^{RR}) with fewer states? NOPE
  - Brzozowski, J.A. Canonical regular expressions and minimal state graphs ford efinite events. In *Proc. Sympos. Math. Theory of Automata (New York, 1962)*, pages 529–561. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963.



Beyond Regular Languages

- Beyond regular languages
  - $a^n b^n$ = {ab, aabb, aaabbb, aaaabbbb, ... } $n \geq 1$
  - is not a regular language

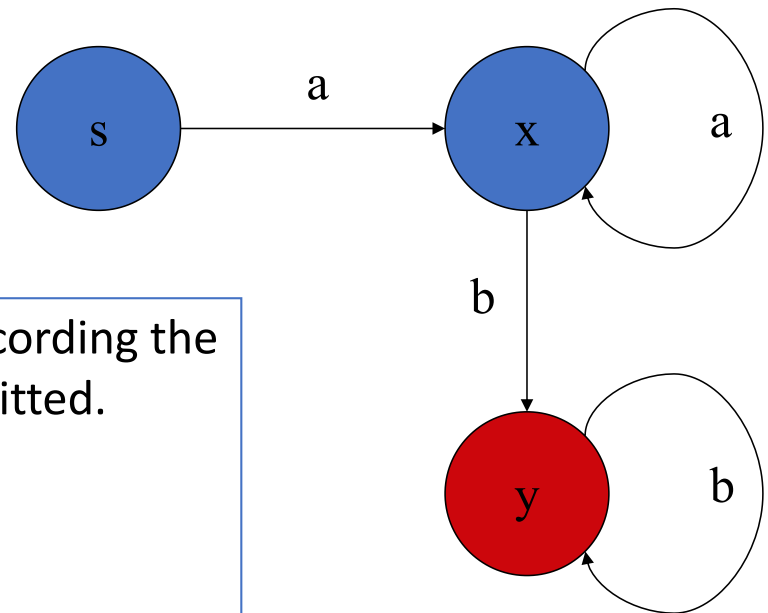
- That means no FSA, regex (or Regular Grammar) can be built for this set

- Informally, let's think about a FSA implementation …

1. We only have a finite number of states to play with …
2. We're only allowed simple free iteration (looping)

Beyond Regular Languages

- L = a^+b^+



Having a frequency table recording the number of visits is not permitted.
Not allowed:

```
%freq = ();

… $freq{$state}++;
```

A Formal Tool: The Pumping Lemma

[See also discussion in JM 16.2.1, pages 533–534]

- Let ***L*** be a regular language,
- then there exists a number $p > 0$
  - where p is a pumping length (*sometimes called a magic number*)

  such that every string w in ***L*** with $|w| \geq p$ can be written in the following form

  w = xyz
- with strings x, y and z such that $|xy| \leq p$, $|y| > 0$ and

  $\boldsymbol{xy^i z}$ is in L
- for every integer $i \geq 0$.

BTW: there is also a pumping lemma for Context-Free Languages

A Formal Tool: The Pumping Lemma

Restated:

- For every (*sufficiently long*) string *w* in a regular language
- there is always a way to split the string into three adjacent sections, call them x, y and z, (y nonempty), i.e. w is x followed by y followed by z
- And y can be repeated as many times as we like (or omitted)
- And the modified string is still a member of the language

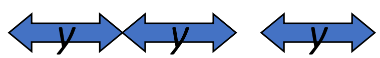
Essential Point!
To prove a language is non-regular: show that no matter how we split the string, there will be modified strings that can't be in the language.

A Formal Tool: The Pumping Lemma

- Example:
  - show that $a^n b^n$ is not regular
- Proof (by contradiction):
  - pick a sufficiently long string in the language
  - e.g. a..aab..bb   (#a's = #b's)
  - Partition it according to ***w = xyz***
  - then show $xy^i z$ is **not** in L
  - i.e. *string does not pump*

A Formal Tool: The Pumping Lemma

aaaa..aabbbb..bb



Case 1: ***w = xyz, y straddles the ab boundary***
what happens when we pump y?

Case 2: ***w = xyz, y is wholly within the a's***
what happens when we pump y?

Case 3: ***w = xyz, y is wholly within the b's***
what happens when we pump y?

A Formal Tool: The Pumping Lemma

- Prime number testing

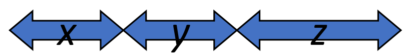
  prime number testing using Perl's extended "regular expressions"

  - Using unary notation, e.g. 5 = "11111"
  - /^(11+?)**\1**+$/ will match anything that's greater than 1 that's not prime

L = $\{1^n \mid n \text{ is prime}\}$ is not a regular language

A Formal Tool: The Pumping Lemma

1^n = 111..1111..11111 such that *n* is a prime number



For any split of the string
Pump *y* such that *i* = length(*x*+*z*), giving y^i

What is the length of string $w = xy^iz$ *now*?

In x y^{xz} z , how many copies of xz do we have?
Answer is y+1
i.e. pumped number can be factorized into (1+|y|)|xz|

i.e., we can show any prime number can be pumped into a non-prime …

The resulting length is non-prime since it can be factorized

A Formal Tool: The Pumping Lemma

$1^n = \underbrace{111..}_{x}\underbrace{1111..}_{y}\underbrace{11111}_{z}$ such that *n* is a prime number

- Illustration of the calculation:

1111 1111 111 (eleven)

1111 1111 1111 1111 1111 1111 1111 1111 111

4 + 4*7 + 3

= 5*7

which isn't prime

- Another look:

1111 111 1111 (re-arrange eleven)

1111 111 1111 1111 1111 1111 111 111 111 111 (make 4 bundles of 4; 4 bundles of 3)

A Formal Tool: The Pumping Lemma

- Another angle to reduce the mystery, let's think in terms of FSA. We know:
  1. we can't control the loops
  2. we are restricted to a finite number of states
  3. assume (without loss of generality) there are no e-transitions

- Suppose there are a total of p states in the machine
- Supose we have a string in the language longer than p
- What can we conclude?

Answer: we must have visited some state(s) more than once!

Also: there must be a loop (or loops) in the machine!

Also: we can repeat or skip that loop and stay inside the language!