

# Regular Expressions

No weekly HW for next week.

(The one after will be slightly longer -

I might announce some problems Friday)

So far, our descriptions of languages have been ad hoc-specific to each case.

Regular expressions are a way of standardizing descriptions of regular languages.

Def: A language is regular if it is the language accepted by some DFA (or equivalently NFA).

- Goals:
- ① Define regular expression (and what language it describes)
  - ② Every reg. expr. describes a reg. lang.
  - ③ Every reg. lang. can be described by a reg. expr.

on  $\Sigma$  recursively

Def: A regular expression is defined by

the following rules

① Regular expr. are strings on  $\Sigma \cup \{\lambda, \emptyset, (, ), +, *\}$

① Single letters in  $\Sigma$  are R.E.

$\lambda$  is an R.E.

$\emptyset$  is an R.E.

② If  $w$  and  $v$  are R.E., then  $wv, w+v, (w), w^*$  are all R.E.'s.

---

What do these mean?

Terminology: A R.E. describes a set of strings (in other words, a language) - a string in the language is said to match the R.E.

Rules:

- If  $l \in \Sigma$ , then the only string matching  $l$  is ' $l$ '.
- ~~The~~ The only string matching  $\lambda$  is the empty string.
- No strings match  $\emptyset$ .

2 a) A string  $u$  matches a R.E.  $wv$  if there is a way of writing  $u = u' u''$  so that  $u'$  matches  $w$  and  $u''$  matches  $v$

b) A string  $u$  matches the R.E.  $w|v$  if  $u$  matches  $w$  or  $u$  matches  $v$ .

c) A string  $u$  matches  $(w)$  if  $u$  matches  $w$ .

d) A string  $u$  matches  $w^*$  if

i)  $u = \lambda$  or

ii)  $u = u' u''$  where  $u'$  matches  $w$  and  $u''$  matches  $w^*$

---

E.g.  $\Sigma = \{a, b\}$

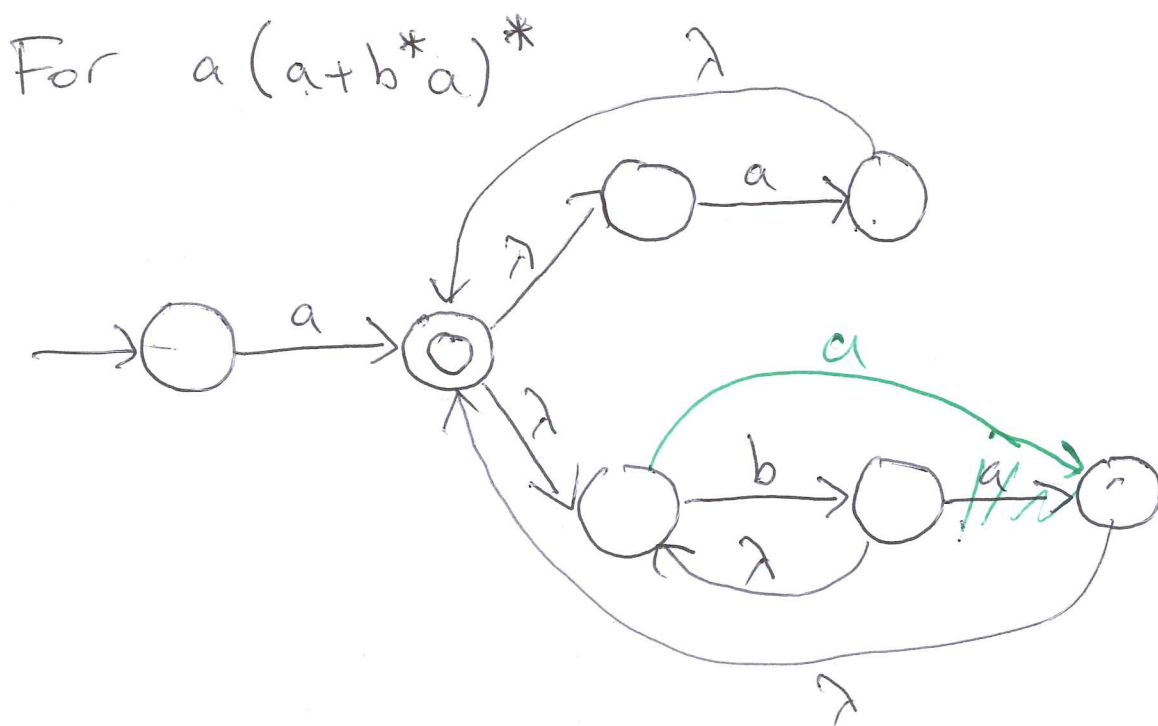
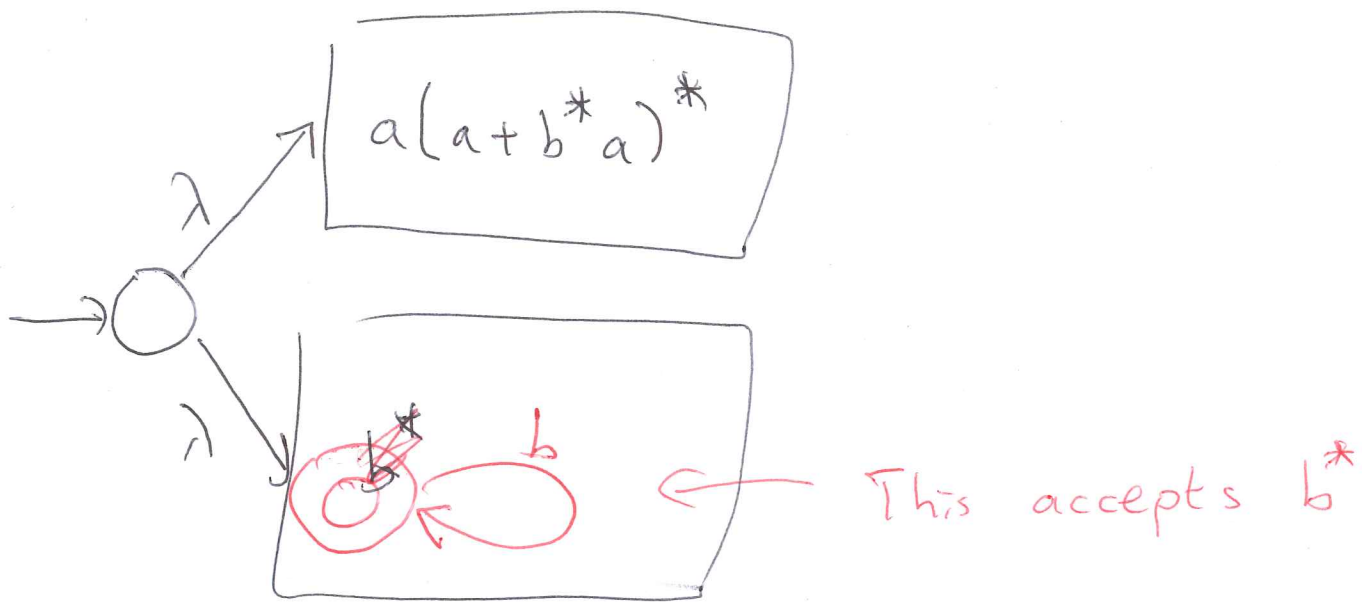
Look at  $a(a + b^* a)^* + b^*$

What strings match this R.E.?

$\lambda, a, b, aa, bb, aaa, aba, bbb, aaaa, abba, bbbb, abaa, aaba$

I'm 90%<sup>?</sup> sure  $\mathbb{R}$  that a string matches this R.E. if it is all b's or it starts and ends w/ an a.

Goal 2: Construct an NFA that accepts a given R.E.



In general:

1a) To accept  $l \in \Sigma$ :



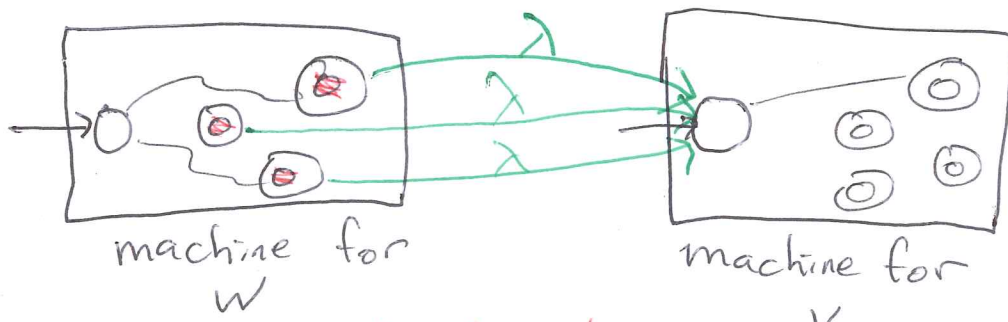
b) To accept  $\lambda$ ,



c) To accept  $\phi$ ,

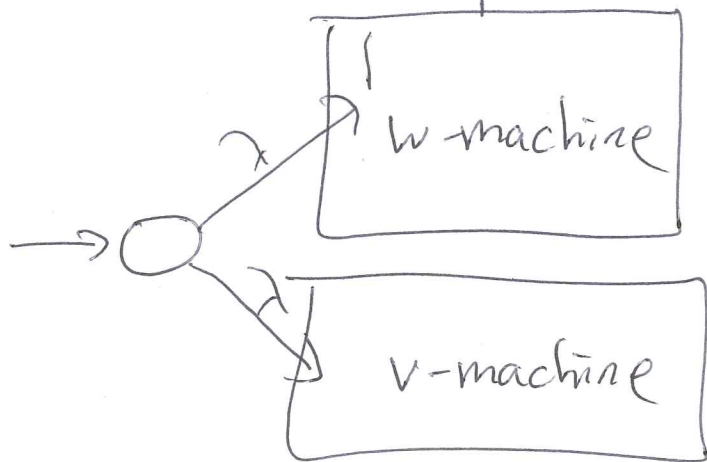


2a) To accept  $wv$



connect final states of w-machine  
to start state of v-machine w/  $\lambda$ -transitions  
Final states of w-machine are no longer  
final states (but final states of v-machine  
~~mach~~ still are)

2b) To accept  $w+v$



2d) To accept  $w^*$

