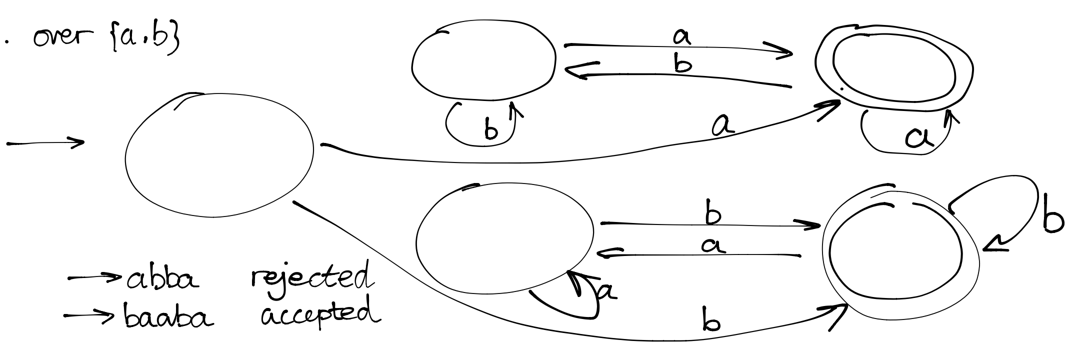
Lecture 19
March 24th, 2015

Formal languages
Alphabet: finite non-empty set of things, we'll call "symbols"
E.g. {a,b}, {0,1,2,3}, {a,b,...,z}

A string over an alphabet Σ is a finite sequence of symbols from Σ.
E.g. Some strings over {a,b}: abba, a, bbb, empty,
Define Σ to be empty string:
A language over alphabet Σ: A set of String over Σ.
E.g. over {a,b}: $L_1 = \{\Sigma, a, aa, aaa, \cdots\}$
$L_2 = \{abba\}$
$L_3 = \{\ \}$

Deterministic
Finite state machine (DFA) (DFSA) over an alphabet Σ.
- Finite A of states
- for each state, a function from Σ to the states
- a subset of the states called the accepting states
- a start state

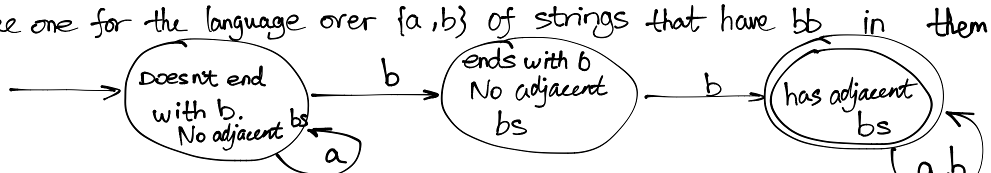
E.g. over {a,b}



→ abba rejected
→ baaba accepted

recognize/accepts the language of string over {a,b} that have odd # of a's

Make one for the language over {a,b} of strings that have bb in them (two adjacent bs)



I(s): S is in state ① iff has even # of a's
∧ in state ② .... odd

a string over is e either · Σ
· S_a or S_b for some string S over {a,b}

Induct on length of string / # of transitions / structure of string
Show for $S = \varepsilon$

ε
a b
aa ba
ab bb

Assume $I(s)$, prove $I(s_a)$, $I(s_b)$:

Case: s has even # a's

$I(s_a)$

$I(s_b)$

Case: s has odd # a's

$I(s_a)$

$I(s_b)$