Language Acceptors

• Let’s augment the FSM, to include a stack of symbols

We’ll store extra information by pushing it down on the stack (hence, “push down” automata)

Stack = primitive memory unit

Often characters we’ll push on, but not limited to characters

Can also pop off of stack
PDA Example

• PDA for recognizing $a^n b^n$

• Informally, push all a’s on, see a b, start popping a’s off. Hope that there aren’t any a’s left when finish with b’s and hope that there are enough a’s to finish with b’s.

• More Formally:
  – FSM component: Need two states to enforce the a-b ordering
  – Use the stack to count the a’s. Match b count to that of a’s by popping off a symbols. First b is a trigger to start popping items off of stack
a^n b^n recognizer PDA

- Let # be end of string and bottom of stack marker (pushed on first)
- ‘read’ outputs are letters consumed from inputs (input alphabet)
- ‘pop’ outputs are letters from stack alphabet (doesn’t have to match input alphabet)