

474 Notes on Day 5 slides:

Some of the early slides are repeats of Day 4; I do not duplicate those notes here.

Slide 14: Finite state machines

Start state, accepting state, dead state.

What can we say about runtime on a string of length N ?

Slide 15: Pushdown Automata

FSM has no memory other than state, so it can't count an arbitrary number of anything.

Input symbol/symbol popped off stack/string pushed on stack.

ϵ for second symbol means "Do this no matter what is on the stack"

ϵ for third symbol means "Don't push anything onto the stack"

BAL = balanced paren language (push/pop parens)

Slide 16: Other examples

Discuss PDA for Bal.

Then have them work on PalSpecial (quiz)

PalEven: Problem is knowing when to push and when to start popping. NONDETERMINISTIC machine.

Slide 17: Trying Another PDA

We'll see later that it is impossible

Slide 26: Nondeterministic computation

First case: Each action will return True, return False, or run forever.

If any of the actions returns TRUE, choose returns true.

If all of the actions return FALSE, choose returns FALSE.

If none of the actions return True, and some do not halt, choose does not halt.

Second case: S may be finite or infinite (with a generator(enumerator)).

If P returns true on some x , so does Choose

If it can be determined that $P(x)$ is false for all x in P , return False.

Otherwise, fail to halt.