

## 474 Notes on Day 6 slides:

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

Slide 3: Trying Anther PDA

We'll see later that it is impossible

Slide 13: 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.

Slide 15: Sample decision problems

A decision problem is a problem that has a yes/no answer for each instance.

A decision problem may be decidable, undecidable, or semidecidable.

A decision procedure is an algorithm to solve a decision problem.

Must always halt and return an answer.

Slide 17: Sample Decision Problems

Prime Fermat Numbers

The first one is clearly decidable: Sketch the algorithm.

What about the second one? Why won't the same approach work?

But there is a decision procedure for this problem:

Return yes

Or

Return No.

One of these is the correct algorithm, but we just don't know which one.

Slide 17: Sample Decision Problems

These turn out not to be decidable.

Slides 18-20 were hidden for 201220

Slide 20: A tractability hierarchy

P contains those languages that can be decided in time that grows as some polynomial function of the length of the input string

NP contains those languages that can be decided by a nondeterministic machine (one that can conduct a search by correctly guessing which move to make) with the property that the amount of time required to explore one sequence of guesses (one path) grows as some polynomial function of the length of the input

PSPACE contains those languages that can be decided by a machine whose space requirement grows as some polynomial function of the length of the input.

Slide 21: Functions on Languages

$maxstring(A^n B^n) = A^n B^n - \{\epsilon\}$

$maxstring(a^*) = \emptyset$

FIN, yes, because every subset of a finite set is finite.

INF, no, see  $\{a\}^*$  example

Slide 22: Functions on Languages

Empty set

$\{a^{2n+1} b^{2n} c^{2n+1}\}$

Yes (FIN)

No (INF)

Slide 23: Functions on Languages

$firstchars(A^n B^n) = \{a\}^*$

$firstchars(\{a, b\}^*) = \{a\}^* \cup \{b\}^*$

FIN, no.  $firstchars(\{a\}) = \{a\}^*$

INF, yes.

Slide 29: Another FSM Example

But it is more complicated, because time is also an input.

But the use of the buttons can be modeled as an FSM.

Slide 36: Accepting and rejecting

201020: We got to this slide but not to the quiz question/proof.

If time, have them do FSM diagram for "divisible by 3"

Base case: If  $w$  is  $\epsilon$ , it halts in 0 steps.

Assume true for strings of length  $n$  and show for strings of length  $n+1$ .

Let  $w \in \Sigma^*$ ,  $w \neq \epsilon$   $|w| = n+1$  for some  $n \in \mathbb{N}$ .

Then  $w$  is  $ax$  for some  $a \in \Sigma$ ,  $x \in \Sigma^*$ ,  $|x| = n$ .

Let  $q'$  be  $\delta(q, a)$ . Then  $(q, w) \vdash_M (q', x)$

By induction, from configuration  $(q', x)$ ,  $M$  halts in  $n$  steps.

So, starting from the original configuration,  $M$  halts in  $n+1$  steps.

