

Defⁿ: A 2-tape TM is a Pushdown Automaton

if in any instruction $(p, c_1 / d_1, D_1, q)$,
 $(p, c_2 / d_2, D_2, q)$,

$D_1 \neq L$ (input tape is one-way, hence *essentially read only*)
 $D_2 = L \Rightarrow d_2 = \square$ (\equiv Tape 2 behaves as a stack with the "top" at right and any left move writing \square means to pop.)

One Tape TM treats its input tape as two tracks.
 Second track uses 0, 1 least sig bit leftmost to hold the current count and it "piggybacks" along.

Keep count in binary of $\#left(x, --)$
 $10 = 2$
 $11 = 3$

On a string Counter never goes higher than n .
 So it uses only $\lceil \log n \rceil$ space at most.
 Each update might need $O(\log n)$ time for "carries"
 \therefore Total time is $O(n \log n)$. *Can be $O(n)$ with a different number repⁿ.*

How a 1-tape machine can do BAL in $O(n \log n)$ time.
 A different way a 2-tape TM can do it in $O(n \log n)$ time or better?

The next part of the lecture was a demo from the "Turing Kit" software of a deterministic pushdown automaton (DPDA) that recognizes the language BAL and also runs in linear time. Then the lecture went on to 1-tape and 2-tape TMs that recognize the language in other ways:

treats its tracks

0, 1 least sig bit leftmost
 count $10 = 2$
 piggybacks along

Keep count in binary of $\#left(x, --x_i) - \#right(x, -x_i)$

On a string x of length n , Counter never goes higher than n , So it uses only $\lceil \log n \rceil$ space at most.
 Each update might need $O(\log n)$ time for "carries"
 \therefore Total time is $O(n \log n)$. *Can be $O(n)$ with a different number repⁿ.*

Informal Defⁿ of "Streaming Alg^m"

- Input tape is one-way, Second tape(s) use at most $O(\log n)$ [or $\log n^{O(1)}$] cells.
- Input tape is allowed $O(\log n)$ lag, where it can go left for $\log n$ steps but then must resume $\log n$ right. Time is $O(n \log n)$, **cannot be improved.**
- Allow $O(\log n)$ repeated passes on output from the previous pass. Every pass halves the length, so $O(\log n)$ passes.

Accept iff you finally get $()$ or ϵ .