Today: *Alternation* vs. *Time* vs. *Space*

- *Fortnow's Theorem:*
  \[ \text{SAT} \in \bigcup \text{Time}(n \log^c n) \]
  \[ \text{OR SAT} \in \text{L} \]

---

**FIRST: A Clarification**

- **What is** *Time* \((f(n))\) ?

  **Answer:** Problems solvable by Deterministic TM with \(\text{finite} \) \# tapes/heads in time \(f(n)\).

\[ \text{finite} = 1 \]
So \( \text{PALINDROME} \in \text{TIME}(O(n)) \)!

(while problem 6 in pset 2 expects you to show that 1-tape TM can't solve problem in time \( n^2 \))

\[ \times \]

On problem 1 of pset 1: Everyone assumed 1 tape & showed asserted

\[ \text{TIME}(O(f(n))) \neq \text{TIME}(f(n) \log f(n)) \]

Claim 1: for 1 tape m/c don't need \( \log f(n) \) extra time

Extra Credit Exercise

Claim 2: Above assertion holds for multitape m/c. Check this

Reading Exercise
ALTERNATION:

ATM: consists of a Turing Machine with 2 special states

- Machine entering $E$ state accepts if one of the outgoing paths accepts.
- Machine entering $A$ state accepts if both outgoing paths accept.

(Natural extension of non-determinism)
Computation of ATM

Represented by a huge tree, with binary branching for \( E \) and \( A \) nodes.
Resources

- **Time** = Depth of tree
- **Space** = as usual
- **Alternations** = # switches from \(\mathcal{E}\) state to \(\mathcal{A}\) state in tree (max over paths) + 1.

![Diagram of a tree showing alternations]

1 alternation

3 alternations.
Why study alternating?

1. Models interesting problems

$$
\text{MIN Alternatives} = \{ (\phi, k) \mid \exists \psi \quad 1\psi \leq k \\
\text{and} \quad \forall x \quad (\phi(x) = \psi(x))
$$

$$
\text{TQBF} = \{ \phi \mid \exists x_1, \forall x_2 \ldots Q_{n_1} x_n \\
\phi(x_1 \ldots x_n) = \text{true} \}
$$

2. Sheds light about Time & Space

[For Now]
Classical relationships

Theorem: \( \text{SPACE}(s) \subseteq \text{ATIME}(s^2) \subseteq \text{SPACE}(s^2) \)

Is this familiar? Replace \( \text{ATIME} \) with \( \text{NSPACE} \) ... !

Proof:

- \( \text{ATIME}(s^2) \subseteq \text{SPACE}(s^2) \): Immediate since \( \text{ATIME}(s^2) \) given by depth \( s^2 \) computation tree can explore entire tree with space \( s^2 \).

- \( \text{SPACE}(s) \subseteq \text{ATIME}(s^2) \)

\[ \text{REACH}(C_1, C_2, 2^s) = \exists C_3 \text{ s.t. } \text{REACH}(C_1, C_3, 2^{s-1}) \]

\[ \supseteq \text{AND } \text{REACH}(C_3, C_2, 2^{s-1}) \]

Then \( s \) space to write ...
Theorem: $\text{TIME}(2^s) \leq \text{ASPACE}(0(s)) \leq \text{TIME}(2^{O(s)})$

Proof:
- Second containment: \# configurations $= 2^{O(s)}$.
- If graph on configurations $= \text{DFA}$, then can determine whether start node accepts or not.
- If graph $= \text{DFA}$, divide into strongly connected components;
  - if state in cycle is rejecting;
  - if state accepts if it leads to accepting state
  
  etc.
- Can determine accept/reject nature of each configuration.
First containment:

To determine if content of DTM tableau in cell \((i,i) = \sigma\) can write \(\text{ATM}\) as follows:

\[
\text{CELL}(i, \delta, \sigma) :=
\begin{cases}
\forall \delta \in \{-1, 0, 1\} \\
\text{CELL}(i - d, j - 1, \sigma_i)
\end{cases}
\]

And \(\sigma_1 \land \sigma_2 \land \sigma_3\) valid for \(\tau\)
**Fortnow's Theorem**

**Idea:** Assume $\text{SAT} \in \text{LIN} \subseteq \bigcup_{c} \text{lin}_{c}$

$\Rightarrow$ $\text{SAT} \in L$

- Then alternations are not useful

\[
\text{(since } \exists L_{1} \subseteq \text{LIN} \subseteq L_{2} \text{)}
\]

- But for small space computation, alternation is powerful; few alternations can reduce time significantly.

- $\text{SAT} \in L \Rightarrow$ all computation is small space.

- So all computation can be speeded up!
DETAILS: Assume \( \text{SAT \in \text{SPACE}}(c \cdot \log n) \) and \( \text{SAT \in \text{TIME}}(n^{1+c}) \)

STEP 1:

1. \( \text{TIME}(T(n)) \subseteq \text{SPACE}(c \cdot \log T(n)) \)
   (needs a strong Cook's Theorem)

STEP 2

2. \( \text{SPACE}(S) \subseteq \text{ATIME}[a, a \cdot 2^{S/a}, S] \)
   \[
   \uparrow \quad \uparrow \quad \uparrow
   \]
   \# alternations \quad \text{time}

1 + 2 \Rightarrow 3:

\( \text{TIME}[T(n)] \subseteq \text{ATIME}[a, a \cdot (T(n)) \cdot \log T(n)] \)
STEP 3

(4) \( A_{\text{TIME}}[a, t] \subseteq T_{\text{TIME}} \left[ t^{(1+\epsilon)^a} \right] \)

(Induction on \( a \); Cook's Theorem)

(3) + (4): (5)

\[ \text{TIME} (T(n)) \subseteq T_{\text{TIME}} \left( T(n)^{\frac{C}{a} \cdot (1+\epsilon)^a} \right) \]

Contradiction if

\[ \frac{C}{a} (1+\epsilon)^a < 1 \]

\[ \forall C < 0 \ \exists \epsilon > 0 \ \text{s.t.} \]

\[ \text{SAT} \in \text{SPACE} \left[ C \cdot \log n \right] \]

\( \Rightarrow \) \text{SAT} \in \text{TIME} \left( n^{1+\epsilon} \right) \)