CS748: Arithmetic Circuit Complexity.

- Classically, computation is modelled using Turing machines.

- I.e. A computer program is seen as a machine $M = (\Gamma, Q, \delta)$ where,
  - Γ is the alphabet, say $\triangleright$ (start), $\square$ (blank), 0 & 1.
  - Q is the set of states (at least q_s & q_f).
  - δ is the transition function $\delta : Q \times \Gamma^2 \to Q \times \Gamma^2 \times \{S, L, R\}^2$.

    head movement

- Example configuration:

- <u>Time</u> is the number of transition steps.

- <u>Space</u> is the number of work tape cells used.

- For input size n & a function $f : \mathbb{N} \to \mathbb{R}_{>0}$ we can talk about complexity classes $\mathrm{Dtime}(f(n))$ & $\mathrm{Space}(f(n))$ as the set of problems that are computable in time $O(f(n))$ & space $O(f(n))$ respectively.

- This leads us to a <u>zoo</u> of classes!

$$P := \bigcup_{c \in \mathbb{N}} \mathrm{Dtime}(n^c)$$

$$\mathrm{Pspace} := \bigcup_{c \in \mathbb{N}} \mathrm{Space}(n^c)$$

$$NP := \bigcup_{c \in \mathbb{N}} Ntime(n^c)$$

$$\mathbb{L} := Space(\lg n)$$

▷ $\mathbb{L} \subseteq P \subseteq NP \subseteq Pspace \subseteq EXP$
$\subseteq EXPspace \subseteq EEXP.$

- There are also randomized versions:

$$ZPP \subseteq RP \subseteq BPP \subseteq PP \subseteq Pspace$$

- and oracle-based classes:

$$\begin{array}{ccccccccc} \Sigma_1 & \subseteq & \Sigma_2 & \subseteq & \Sigma_3 & \subseteq \ldots\ldots \subseteq & PH & \subseteq & Pspace. \\ \| & & \| & & \| & & \| & & \\ NP & & NP^{\Sigma_1} & & NP^{\Sigma_2} & & \bigcup_{c \in \mathbb{N}} \Sigma_c & & \end{array}$$

- This course will take a different route to build a zoo of computational classes!

- Instead of seeing computation as a sequence of very simple steps, we will view it as an algebraic expression.

- Definition: An arithmetic circuit C, over a field $\mathbb{F}$, is a rooted dag as follows. The leaves are the variables $x_1, \ldots, x_n$ (input) & the root outputs a polynomial $C(\bar{x})$.

  The internal vertices are gates that compute $*$ or $+$ in $\mathbb{F}[\bar{x}]$.

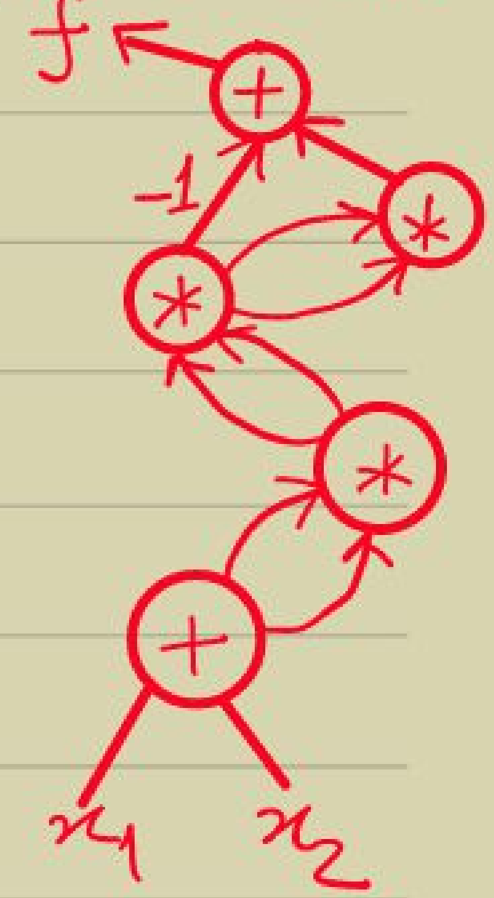
  The edges are called wires & they can have constant (in $\mathbb{F}$) labels to do scalar multiplication.

  The #wires & the size of the constants comprise the size of the circuit C.

  A max-path from a leaf to the root determines the depth of C.

  $\deg(C)$ refers to the degree of the intermediate polynomials computed.

- Eg. The polynomial $f = (x_1+x_2)^8 - (x_1+x_2)^4$ has the following circuit representation:



- Note that the circuit for f is quite compact (though f has 14 monomials!)

- Repeated-squaring is used.

- Definition: fanin (resp. fanout) of a circuit refers to the max indegree (resp. outdegree) of the gates/vertices.
  A circuit with fanout $= 1$ is called a formula.

- Suppose $\mathcal{F} := \{ f_i(x_1, \ldots, x_i) \mid i \in \mathbb{N} \}$ is a family of polynomials (call it problem).
  We will say that a family of