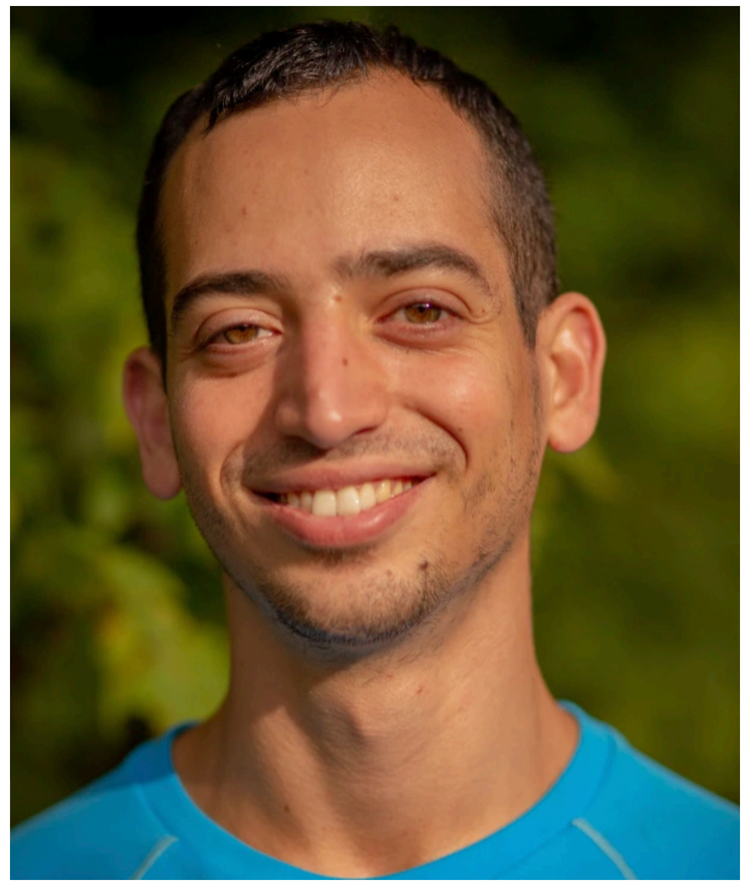
Online Learning and Solving Infinite Games with an ERM Oracle

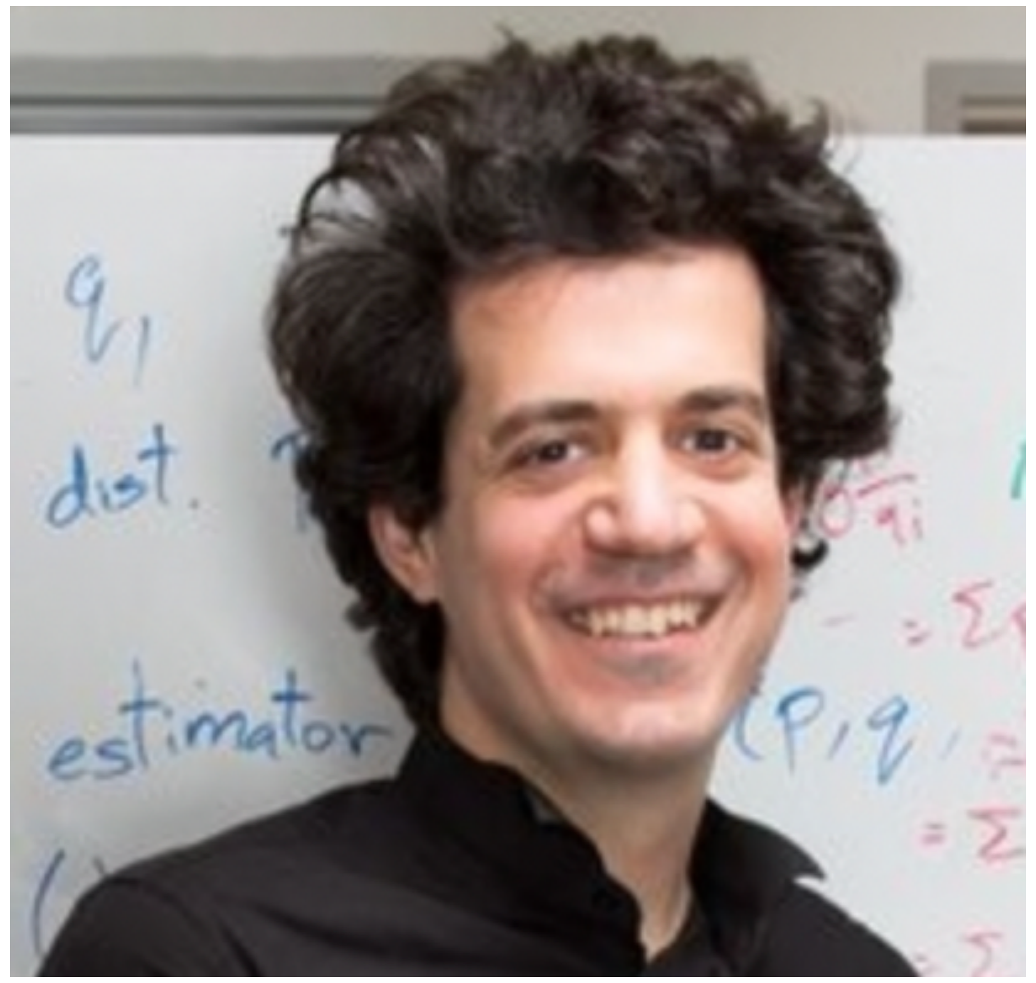
Presentation by Maxwell Fishelson

Angelos
Assos

Idan
Attias



Yuval
Dagan



Costis
Daskalakis

Online Learning

- Data sequence $(x_1, y_1), (x_2, y_2), \cdots$ $x_t \in \mathcal{X}, y_t \in \{0,1\}$
- At each time $t = 1,2,\cdots,T$, learner chooses $h_t : \mathcal{X} \to \{0,1\}$
- Then, (x_t, y_t) revealed
- **Goal:** $h_t(x_t) = y_t$
- Given: hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$

Online Learning

- Data sequence $(x_1, y_1), (x_2, y_2), \cdots$ $\quad x_t \in \mathcal{X}, y_t \in \{0,1\}$
- At each time $t = 1,2,\cdots,T$, learner chooses $h_t : \mathcal{X} \to \{0,1\}$
- Then, (x_t, y_t) revealed
- **Goal:** $h_t(x_t) = y_t$
- Given: hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$

- **Realizable:** assume $\exists h \in \mathcal{H}$ with $h(x_t) = y_t$ for all t
- Want: mistake bound M
- $\sum_{t=1}^{T} 1[h_t(x_t) \neq y_t] \leq M$

Online Learning

- Data sequence $(x_1, y_1), (x_2, y_2), \cdots$ $\quad x_t \in \mathcal{X}, y_t \in \{0,1\}$
- At each time $t = 1,2,\cdots, T$, learner chooses $h_t : \mathcal{X} \to \{0,1\}$
- Then, (x_t, y_t) revealed
- **Goal:** $h_t(x_t) = y_t$
- Given: hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$

- **Realizable:** assume $\exists h \in \mathcal{H}$ with $h(x_t) = y_t$ for all t
- Want: mistake bound M
- $\sum_{t=1}^{T} 1[h_t(x_t) \neq y_t] \leq M$

- **Agnostic:** arbitrary sequence $\{(x_t, y_t)\}_{t=1}^{T}$
- Want: regret bound R_T
- $\sum_{t=1}^{T} 1[h_t(x_t) \neq y_t] - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} 1[h(x_t) \neq y_t] \leq R_T$

General Online Learning

General Online Learning

Want: fewest assumptions possible

General Online Learning

Want: fewest assumptions possible

$\mathscr{H}$ finite: sufficient for online learning, but not necessary

General Online Learning

Want: fewest assumptions possible

$\mathscr{H}$ finite: sufficient for online learning, but not necessary

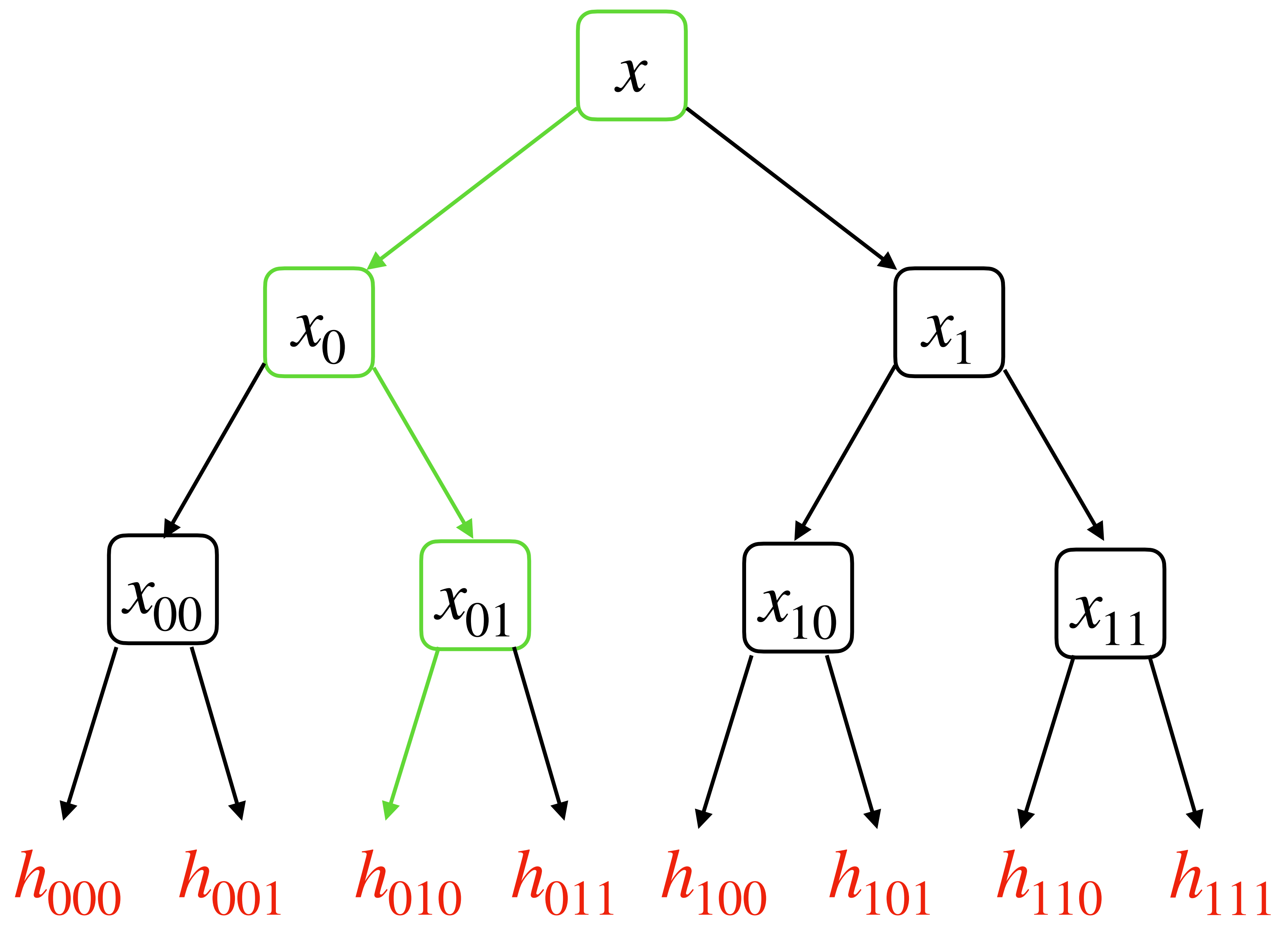
Necessary and sufficient: $\mathscr{H}$ has finite **Littlestone Dimension** [Littlestone 88]

General Online Learning

Want: fewest assumptions possible

$\mathscr{H}$ finite: sufficient for online learning, but not necessary

Necessary and sufficient: $\mathscr{H}$ has finite **Littlestone Dimension** [Littlestone 88]



Example: $h_{010}(x), h_{010}(x_0), h_{010}(x_{01}) = 0,1,0$

General Online Learning

General Online Learning

Assume $\mathrm{Lit}(\mathscr{H})$ finite

General Online Learning

Assume $\text{Lit}(\mathscr{H})$ finite

$\mathscr{H}$ potentially infinite: need **oracle assumption**

General Online Learning

Assume $\mathrm{Lit}(\mathscr{H})$ finite

$\mathscr{H}$ potentially infinite: need **oracle assumption**

Past work: **SOA oracle**

General Online Learning

Assume $\mathrm{Lit}(\mathscr{H})$ finite

$\mathscr{H}$ potentially infinite: need **oracle assumption**

Past work: **SOA oracle**

This work: **ERM oracle**

Oracle Models (Realizable)

Oracle Models (Realizable)

Empirical Risk Minimization (ERM):

Given $(x_1, y_1), \cdots (x_t, y_t)$: return $h \in \mathscr{H}$ with $h(x_\tau) = y_\tau$ for all τ

Oracle Models (Realizable)

Empirical Risk Minimization (ERM):

Given $(x_1, y_1), \cdots (x_t, y_t)$: return $h \in \mathscr{H}$ with $h(x_\tau) = y_\tau$ for all τ

Standard Optimal Algorithm (SOA):

Given $(x_1, y_1), \cdots (x_t, y_t)$: find all consistent $H^{(t)} \subseteq \mathscr{H}$

Oracle Models (Realizable)

Empirical Risk Minimization (ERM):

Given $(x_1, y_1), \cdots (x_t, y_t)$: return $h \in \mathscr{H}$ with $h(x_\tau) = y_\tau$ for all τ

Standard Optimal Algorithm (SOA):

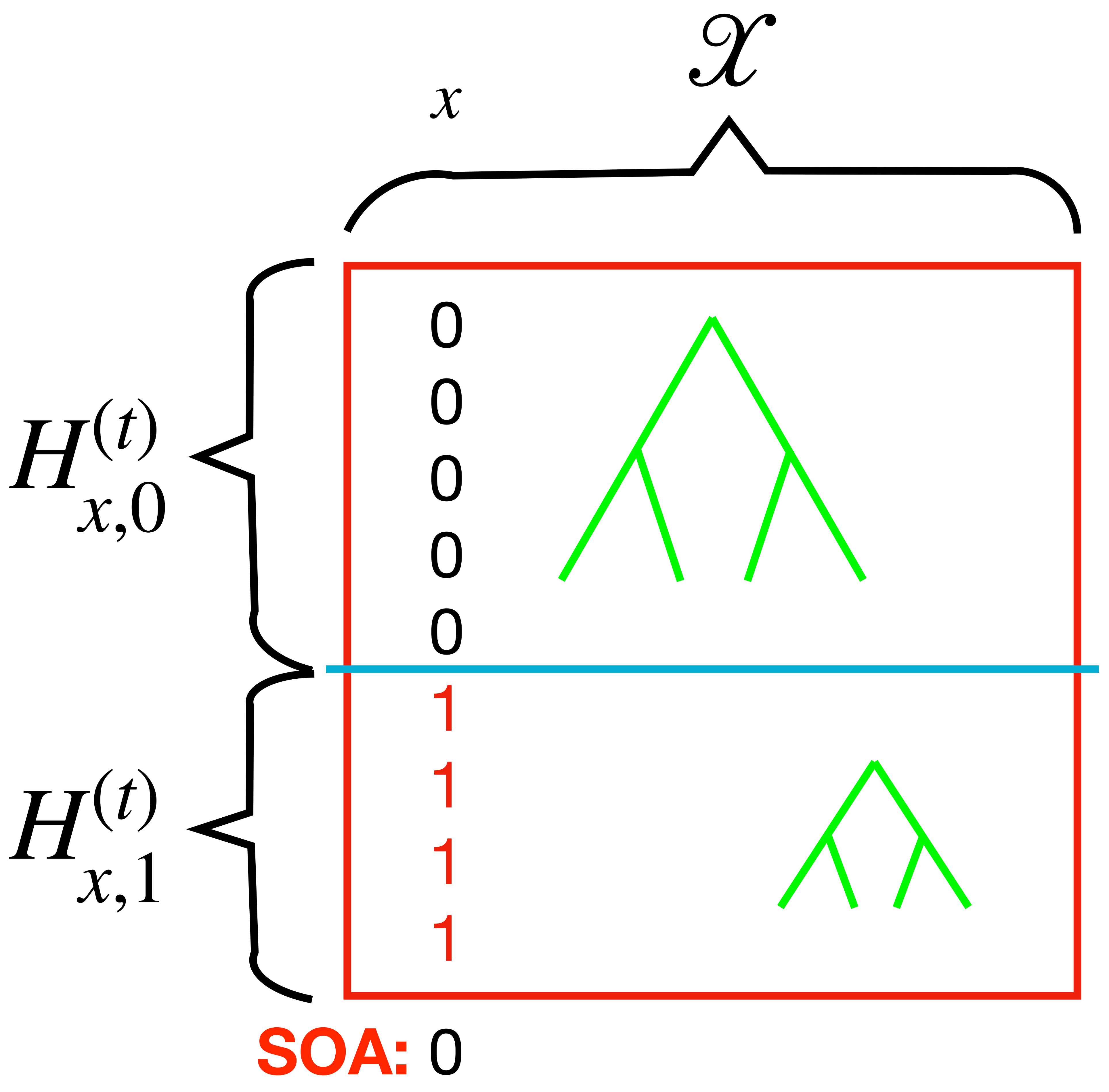
Given $(x_1, y_1), \cdots (x_t, y_t)$: find all consistent $H^{(t)} \subseteq \mathscr{H}$

$H^{(t)}_{x,0} \subseteq \{h \in H^{(t)} \,|\, h(x) = 0\}$

$H^{(t)}_{x,1} \subseteq \{h \in H^{(t)} \,|\, h(x) = 1\}$

Compute Lit: $H^{(t)}_{x,0} \times \mathscr{X}$ and $H^{(t)}_{x,1} \times \mathscr{X}$

Return $\arg\max_y \text{Lit}(H^{(t)}_{x,y} \times \mathscr{X})$ for all x



Why ERM?

SOA Oracle (Strong)	ERM Oracle (Weak)

Why ERM?

SOA Oracle (Strong)	ERM Oracle (Weak)
Cannot distinguish in poly time Lit$(\mathscr{H}) = \tilde{O}(1)$ Lit$(\mathscr{H}) = \tilde{\Omega}(\log \lvert \mathscr{H} \rvert)$ [Manurangsi 22]	

Why ERM?

SOA Oracle (Strong)	ERM Oracle (Weak)
Cannot distinguish in poly time Lit($\mathscr{H}$) $= \tilde{O}(1)$ Lit($\mathscr{H}$) $= \tilde{\Omega}(\log \lvert \mathscr{H} \rvert)$ [Manurangsi 22] **Optimal:** [Littlestone88] [FR98] Realizable: Lit($\mathscr{H}$) mistakes Agnostic: $\sqrt{T \cdot \text{Lit}(\mathscr{H})}$ regret	

Why ERM?

SOA Oracle (Strong)	ERM Oracle (Weak)
Cannot distinguish in poly time Lit$(\mathscr{H}) = \tilde{O}(1)$ Lit$(\mathscr{H}) = \tilde{\Omega}(\log \lvert \mathscr{H} \rvert)$ [Manurangsi 22] **Optimal:** [Littlestone88] [FR98] Realizable: Lit$(\mathscr{H})$ mistakes Agnostic: $\sqrt{T \cdot \text{Lit}(\mathscr{H})}$ regret	• Classic PAC assumption • FTRL, gradient methods, online mirror descent, double-oracle algorithms

Why ERM?

SOA Oracle (Strong)	ERM Oracle (Weak)
Cannot distinguish in poly time Lit($\mathscr{H}$) $= \tilde{O}(1)$ Lit($\mathscr{H}$) $= \tilde{\Omega}(\log \lvert \mathscr{H} \rvert)$ [Manurangsi 22]	• Classic PAC assumption • FTRL, gradient methods, online mirror descent, double-oracle algorithms
Optimal: [Littlestone88] [FR98] Realizable: Lit($\mathscr{H}$) mistakes Agnostic: $\sqrt{T \cdot \text{Lit}(\mathscr{H})}$ regret	**Not Optimal:** Realizable: **finite mistakes?** Agnostic: **sublinear regret?**

Why ERM?

SOA Oracle (Strong)	ERM Oracle (Weak)
Cannot distinguish in poly time Lit($\mathscr{H}$) $= \tilde{O}(1)$ Lit($\mathscr{H}$) $= \tilde{\Omega}(\log \lvert \mathscr{H} \rvert)$ [Manurangsi 22]	• Classic PAC assumption • FTRL, gradient methods, online mirror descent, double-oracle algorithms
Optimal: [Littlestone88] [FR98] Realizable: Lit($\mathscr{H}$) mistakes Agnostic: $\sqrt{T \cdot \text{Lit}(\mathscr{H})}$ regret	**Not Optimal:** Realizable: **finite mistakes?** Agnostic: **sublinear regret?**

Yes! [AADD🐟 23]

Main results (Online Learning)

Regret Bounds	Realizable	Agnostic
Improper: $h_t \in \{0,1\}^{\mathcal{X}}$		

Main results (Online Learning)

Regret Bounds	Realizable	Agnostic
Improper: $h_t \in \{0,1\}^{\mathcal{X}}$		
Proper (randomized): $h_t \in \Delta(\mathcal{H})$		

Main results (Online Learning)

Regret Bounds	Realizable	Agnostic
Improper: $h_t \in \{0,1\}^{\mathcal{X}}$	$C^{Lit(\mathcal{H})}$	$\sqrt{TC^{Lit(\mathcal{H})}}$
Proper (randomized): $h_t \in \Delta(\mathcal{H})$	$T^{\frac{2Lit(\mathcal{H})+2}{2Lit(\mathcal{H})+3}}$	$T^{\frac{2Lit(\mathcal{H})+3}{2Lit(\mathcal{H})+4}}$

Main results (Online Learning)

Regret Bounds	Realizable	Agnostic
Improper: $h_t \in \{0,1\}^{\mathcal{X}}$	$C^{Lit(\mathcal{H})}$	$\sqrt{TC^{Lit(\mathcal{H})}}$
Proper (randomized): $h_t \in \Delta(\mathcal{H})$	$T^{\frac{2Lit(\mathcal{H})+2}{2Lit(\mathcal{H})+3}}$	$T^{\frac{2Lit(\mathcal{H})+3}{2Lit(\mathcal{H})+4}}$

$\mathcal{H}$ class of thresholds: $|\mathcal{H}| = C^{Lit(\mathcal{H})}$ mistakes necessary

Main results (Online Learning)

Regret Bounds	Realizable	Agnostic
Improper: $h_t \in \{0,1\}^{\mathcal{X}}$	$C^{Lit(\mathcal{H})}$	$\sqrt{TC^{Lit(\mathcal{H})}}$
Proper (randomized): $h_t \in \Delta(\mathcal{H})$	$T^{\frac{2Lit(\mathcal{H})+2}{2Lit(\mathcal{H})+3}}$	$T^{\frac{2Lit(\mathcal{H})+3}{2Lit(\mathcal{H})+4}}$

$\mathcal{H}$ class of thresholds: $|\mathcal{H}| = C^{Lit(\mathcal{H})}$ mistakes necessary

SOA: Poly iterations, each iteration exp | **ERM:** Exp iterations, each iteration fast

Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathscr{H}$

Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h \sim \mu^{(t)}}[|h(x_t) - y_t|] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(|h_i(x_i) - y_i|)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$

Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathscr{H}$

Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h \sim \mu^{(t)}}[\,|h(x_t) - y_t|\,] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(\,|h_i(x_i) - y_i|\,)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$

Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathscr{H}$

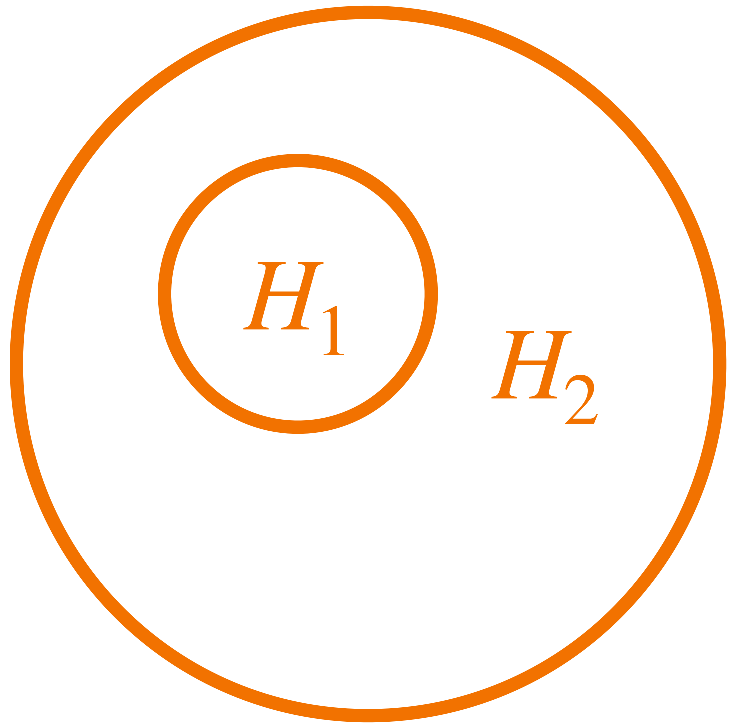
Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h \sim \mu^{(t)}}[\,|h(x_t) - y_t|\,] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(\,|h_i(x_i) - y_i|\,)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$



Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathscr{H}$

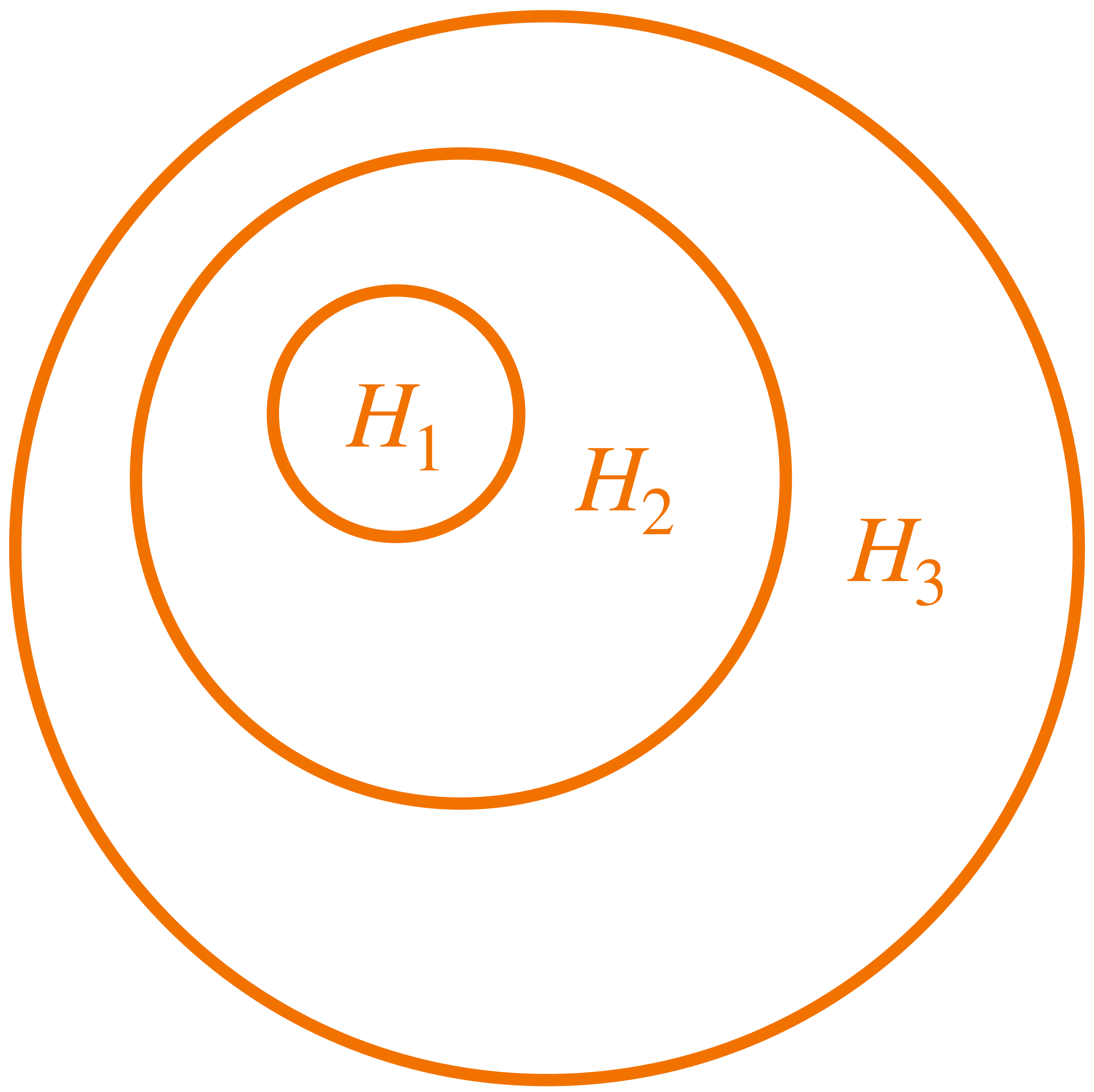
Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h\sim\mu^{(t)}}[\,|h(x_t) - y_t|\,] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(\,|h_i(x_i) - y_i|\,)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$



Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathscr{H}$

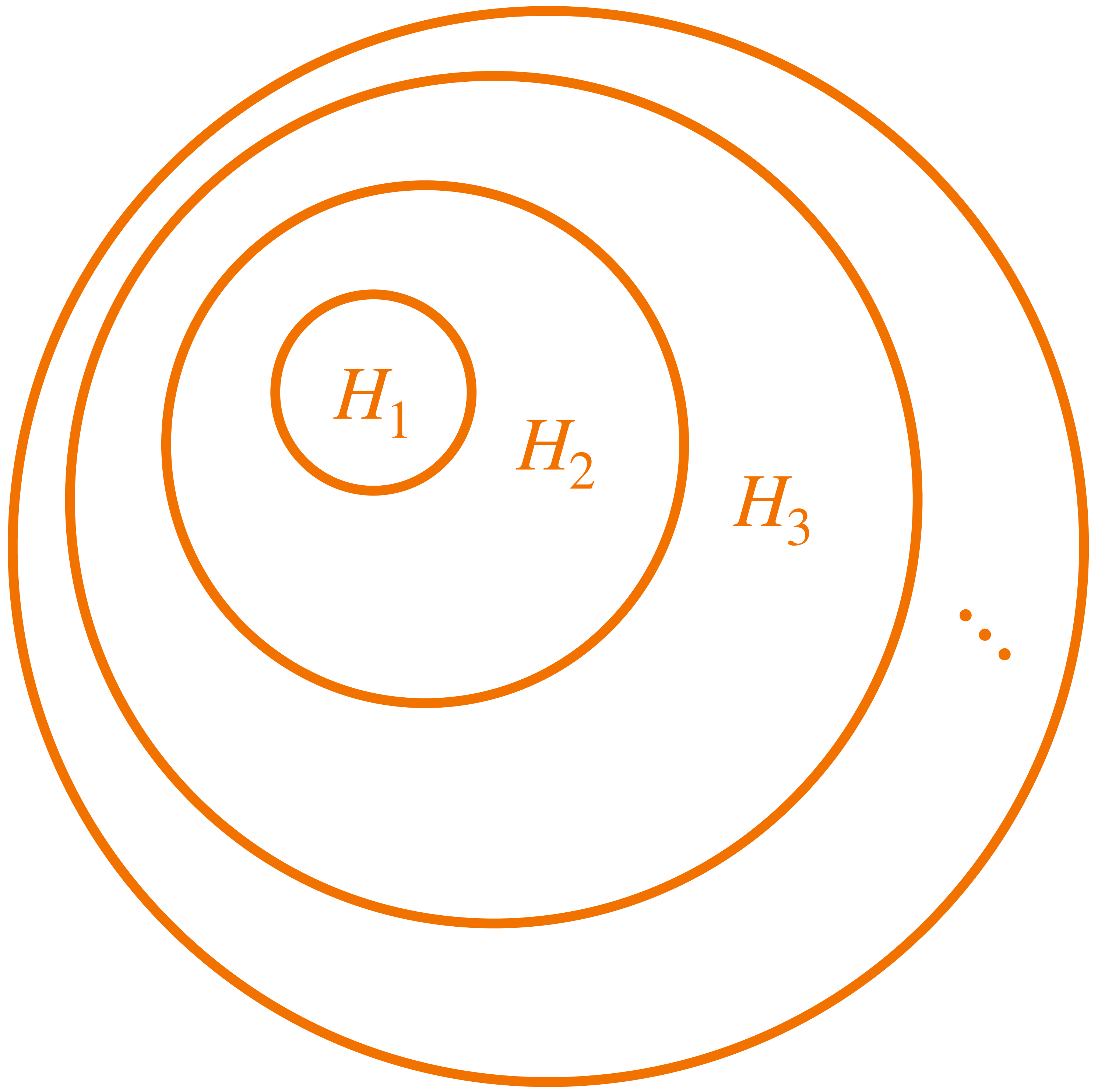
Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h\sim\mu^{(t)}}[\,|h(x_t) - y_t|\,] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(\,|h_i(x_i) - y_i|\,)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$



Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathscr{H}$

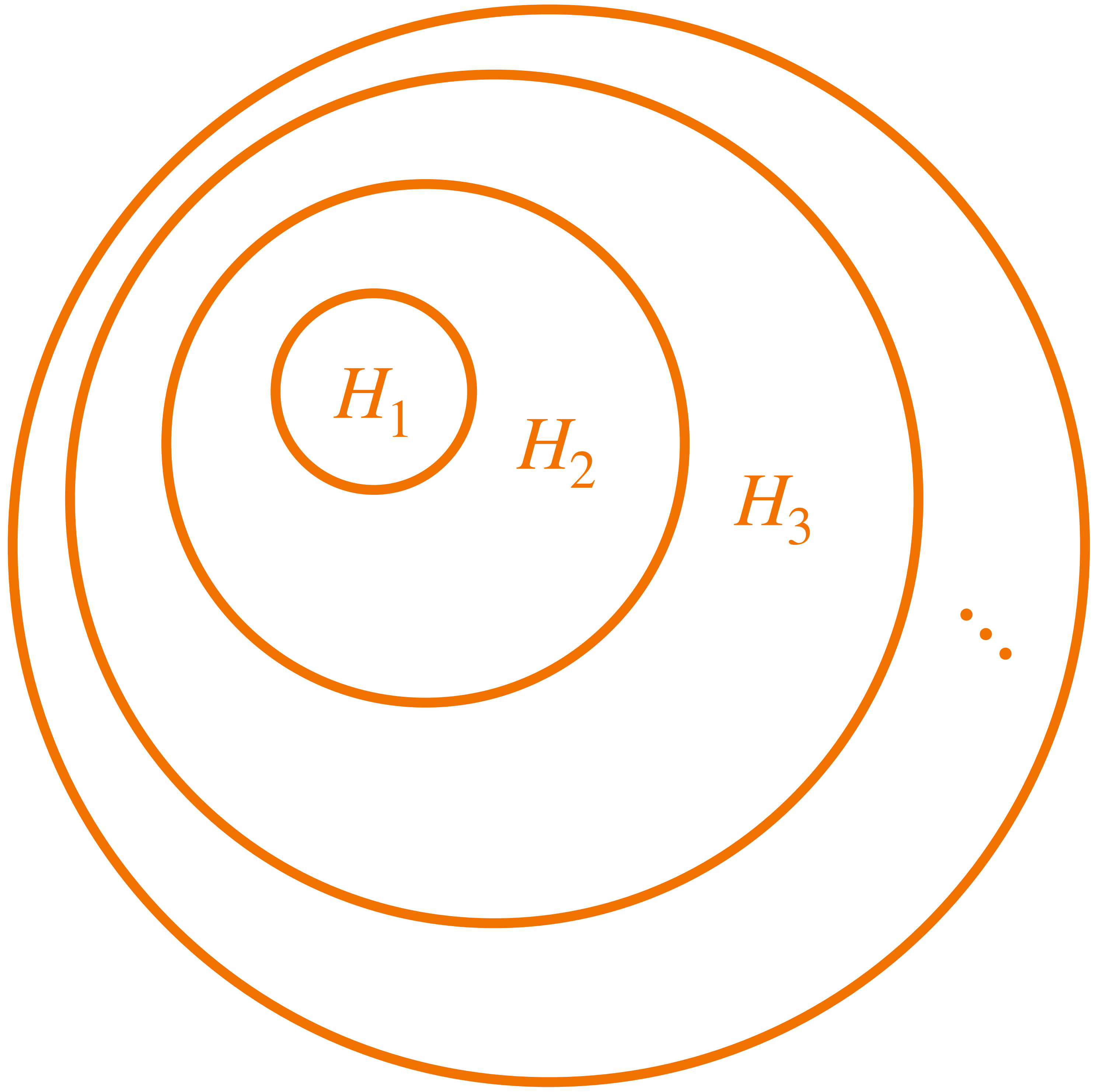
Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h \sim \mu^{(t)}}[\,|h(x_t) - y_t|\,] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(\,|h_i(x_i) - y_i|\,)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$



Proof idea

Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathcal{H}$

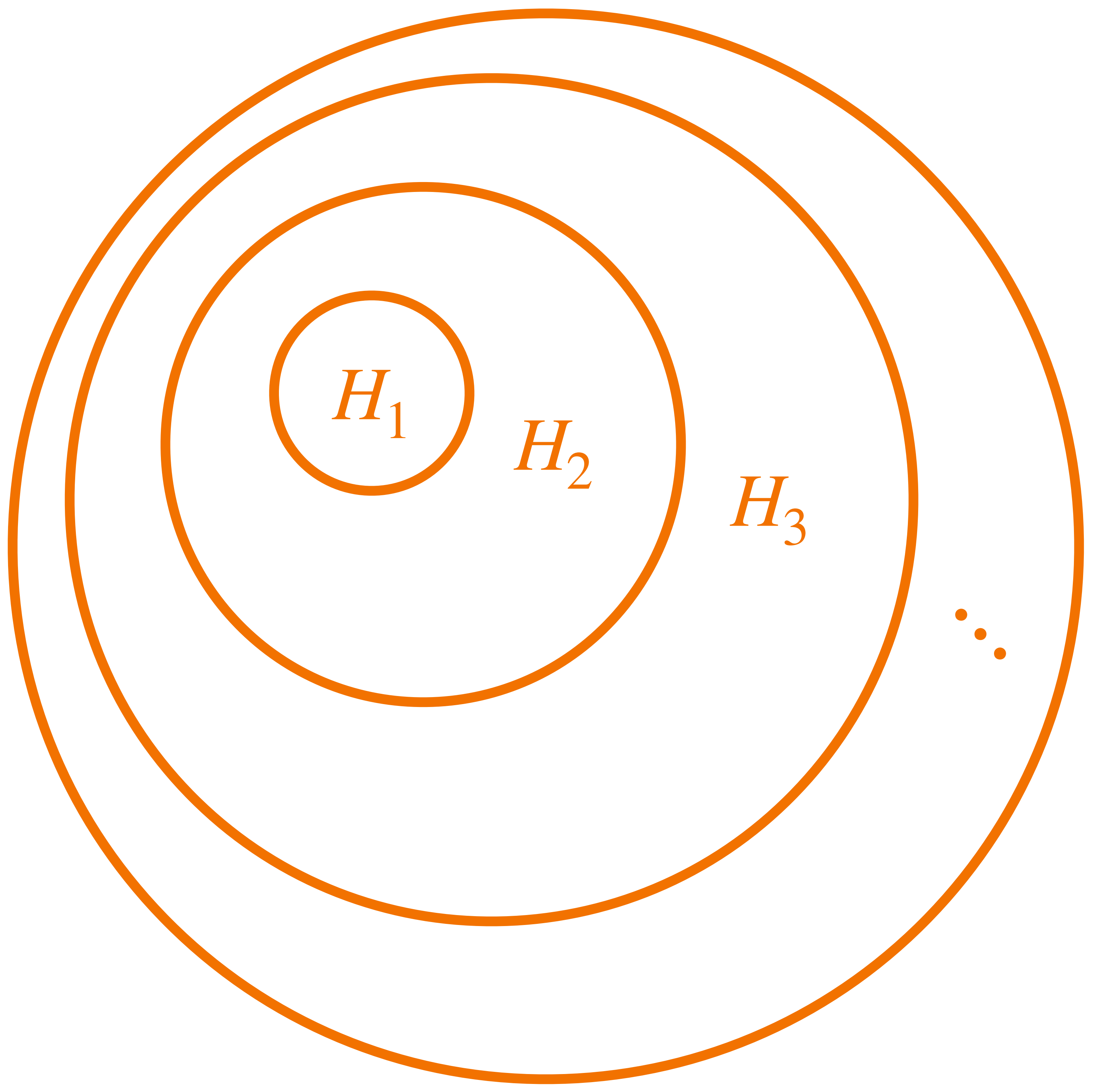
Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h\sim\mu^{(t)}}[\,|h(x_t) - y_t|\,] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(\,|h_i(x_i) - y_i|\,)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$



Proof idea

stops at H_J: roughly $h^* \in H_J$

Algorithm (Realizable Proper)

Stage j: finite $H_j \subseteq \mathscr{H}$

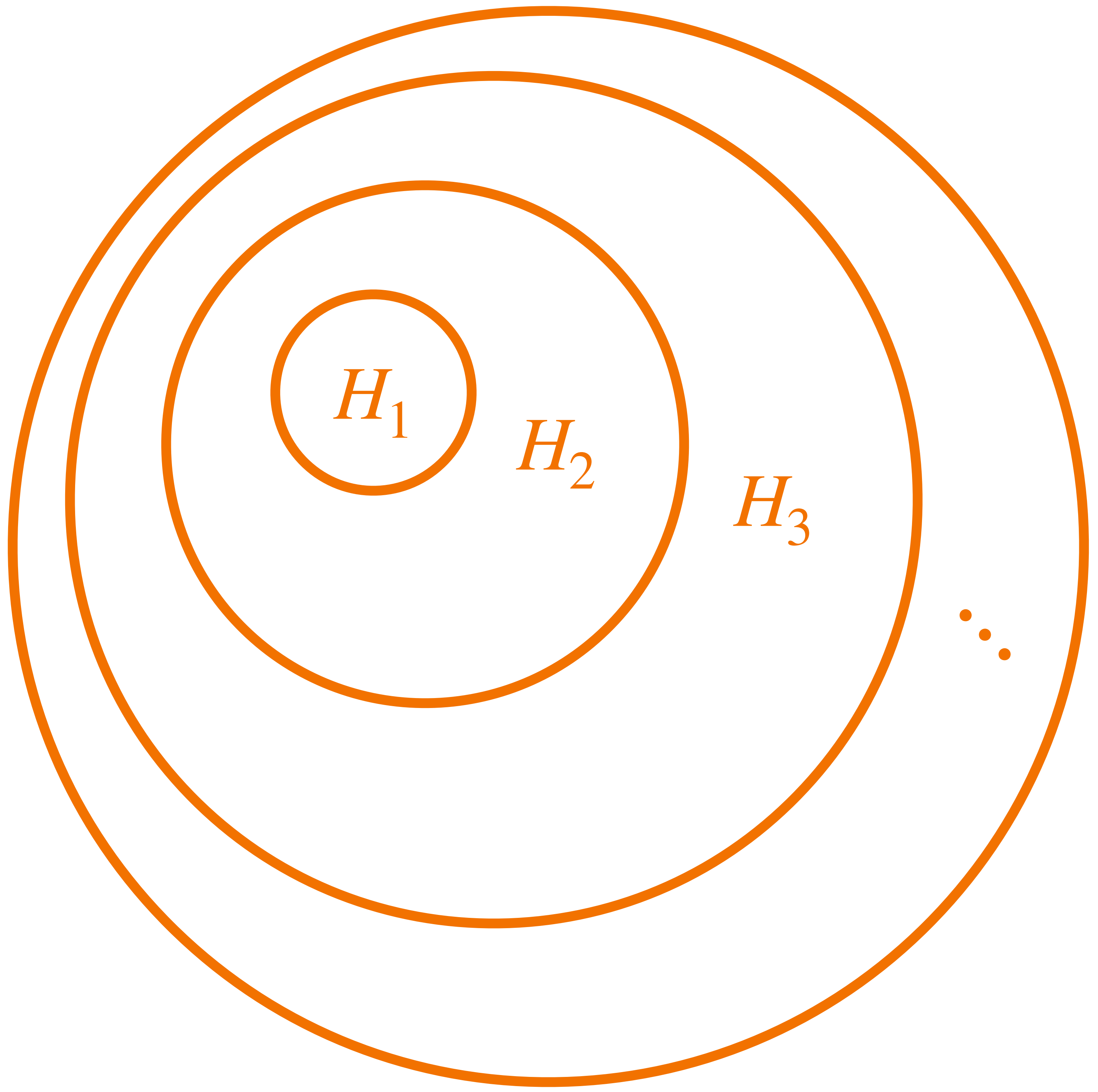
Lazy MWU on H_j: predict $\mu^{(t)} \in \Delta_{H_j}$

If $\mathbb{E}_{h \sim \mu^{(t)}}[\,|h(x_t) - y_t|\,] > \epsilon$, update $\mu_i^{(t+1)} \propto \mu_i^{(t)} \cdot \exp(\,|h_i(x_i) - y_i|\,)$

Otherwise $\mu_i^{(t+1)} = \mu_i^{(t)}$

Stage ends after $O\left(\frac{\log j}{\epsilon^2}\right)$ updates

ERM: $H_{j+1} \leftarrow H_j \cup \{\text{consistent } h\}$



Proof idea

stops at H_J: roughly $h^* \in H_J$

too many stages: construct large Littlestone tree

Main results (Games)

Main results (Games)

Two-player zero-sum game

Action sets: $\mathscr{A}, \mathscr{B}$

Utility $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$

Player 1 chooses $a \in \mathscr{A}$

Player 2 chooses $b \in \mathscr{B}$

Receive reward/loss $u(a, b)$

ϵ-minmax: $\mu_a \in \Delta_{\mathscr{A}}, \mu_b \in \Delta_{\mathscr{B}}$

$$u(\mu_a, \mu_b) \geq \max_{a \in \mathscr{A}} u(a, \mu_b) - \epsilon$$

$$u(\mu_a, \mu_b) \leq \min_{b \in \mathscr{B}} u(\mu_a, b) + \epsilon$$

Main results (Games)

Two-player zero-sum game

Action sets: $\mathscr{A}, \mathscr{B}$

Utility $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$

Player 1 chooses $a \in \mathscr{A}$

Player 2 chooses $b \in \mathscr{B}$

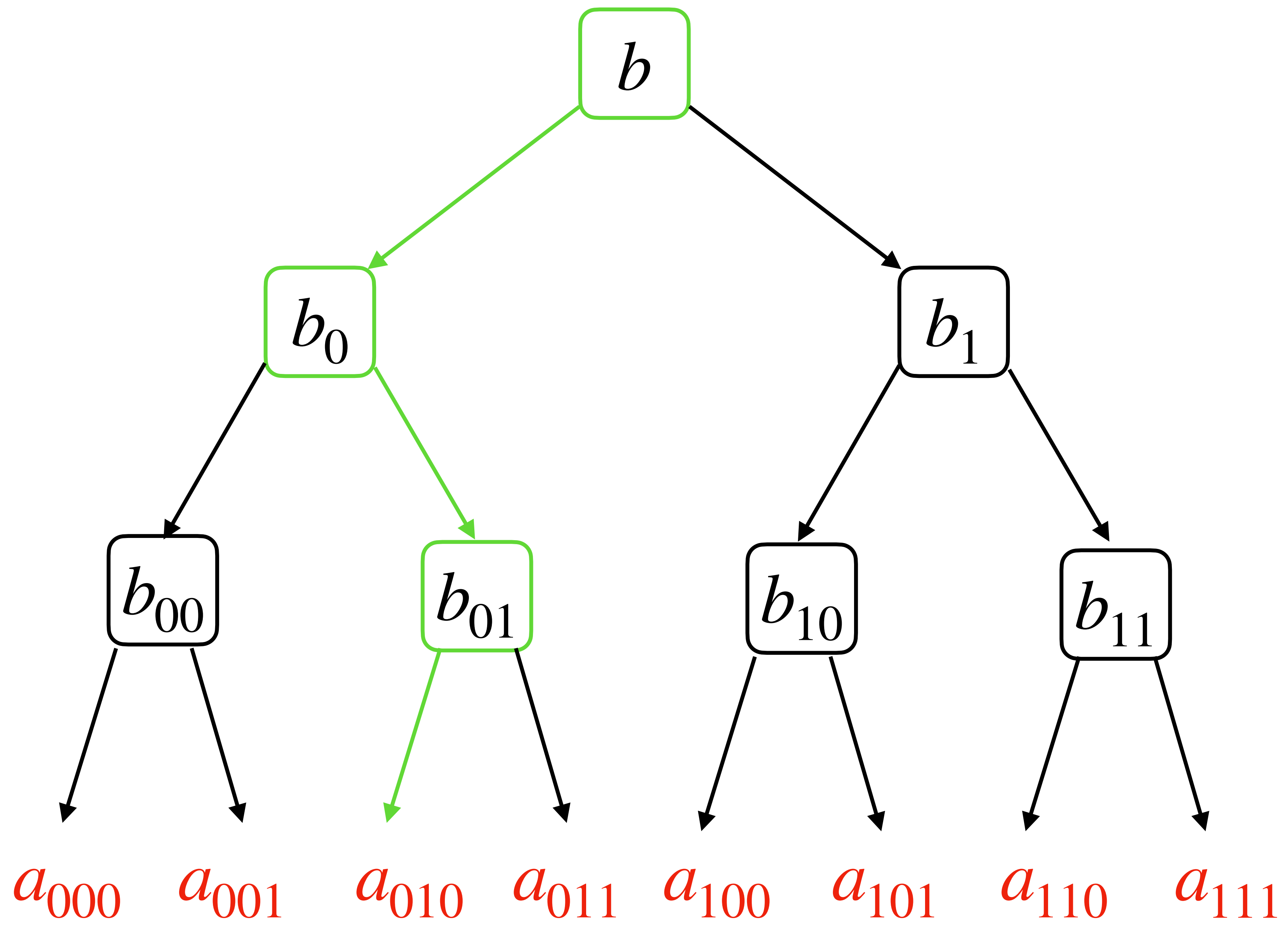
Receive reward/loss $u(a, b)$

ϵ-minmax: $\mu_a \in \Delta_{\mathscr{A}}, \mu_b \in \Delta_{\mathscr{B}}$

$$u(\mu_a, \mu_b) \geq \max_{a \in \mathscr{A}} u(a, \mu_b) - \epsilon$$

$$u(\mu_a, \mu_b) \leq \min_{b \in \mathscr{B}} u(\mu_a, b) + \epsilon$$

Lit(u)



Example: $u(a_{010}, b), u(a_{010}, b_0), u(a_{010}, b_{01}) = 0,1,0$

Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \rightarrow \{0,1\}$, finite $\mathrm{Lit}(u)$

Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite $\mathrm{Lit}(u)$

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$

Algorithm (Two Player Zero Sum Games)

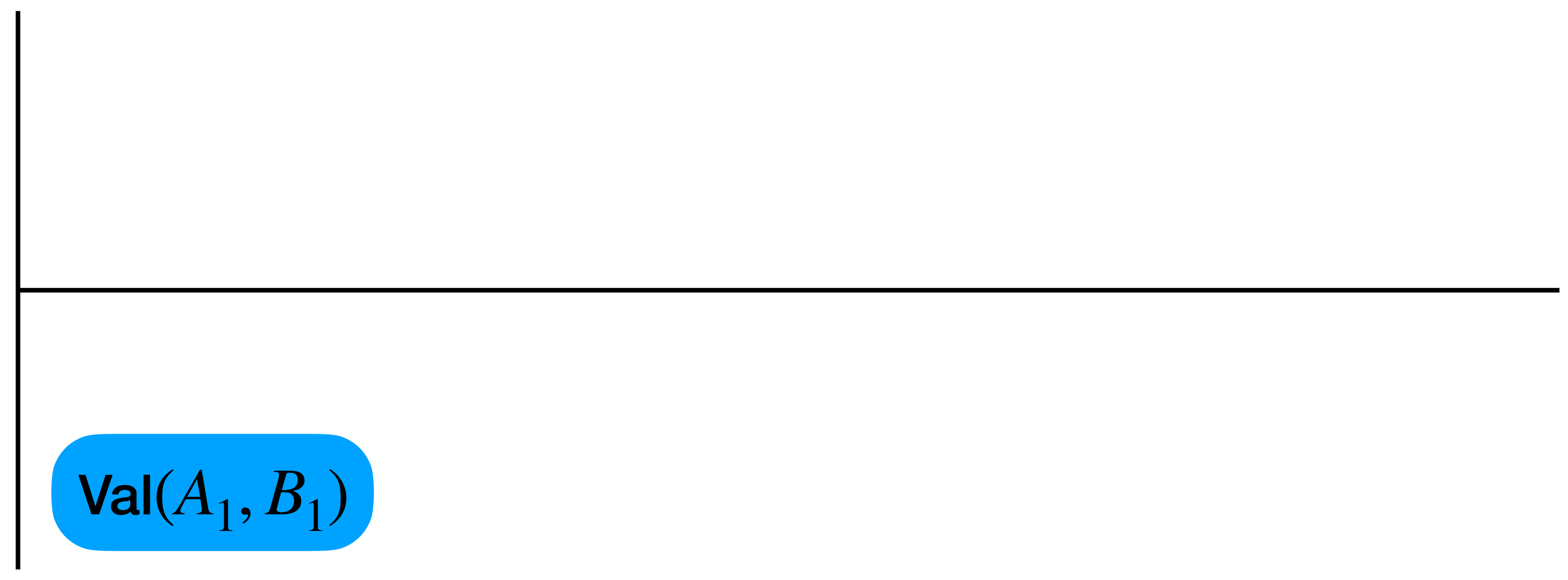
Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite $\mathrm{Lit}(u)$

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$

Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite $\mathrm{Lit}(u)$

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$



Algorithm (Two Player Zero Sum Games)

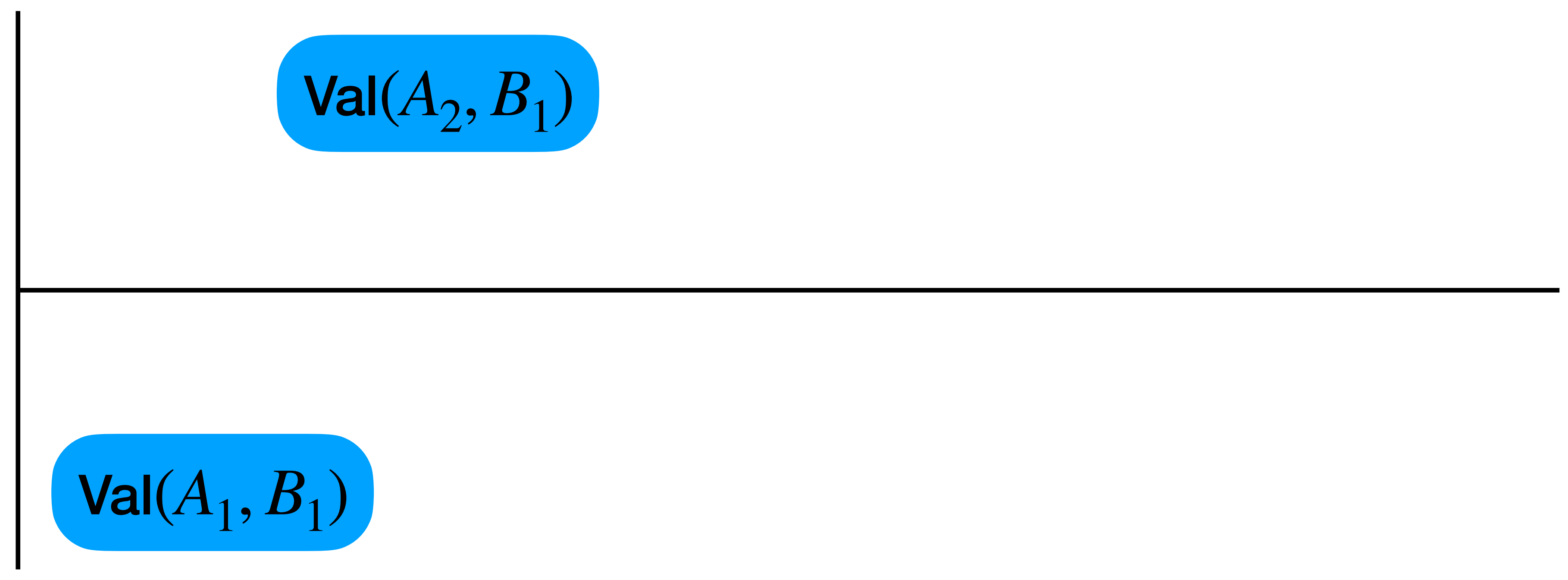
Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite Lit($u$)

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$

Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite Lit$(u)$

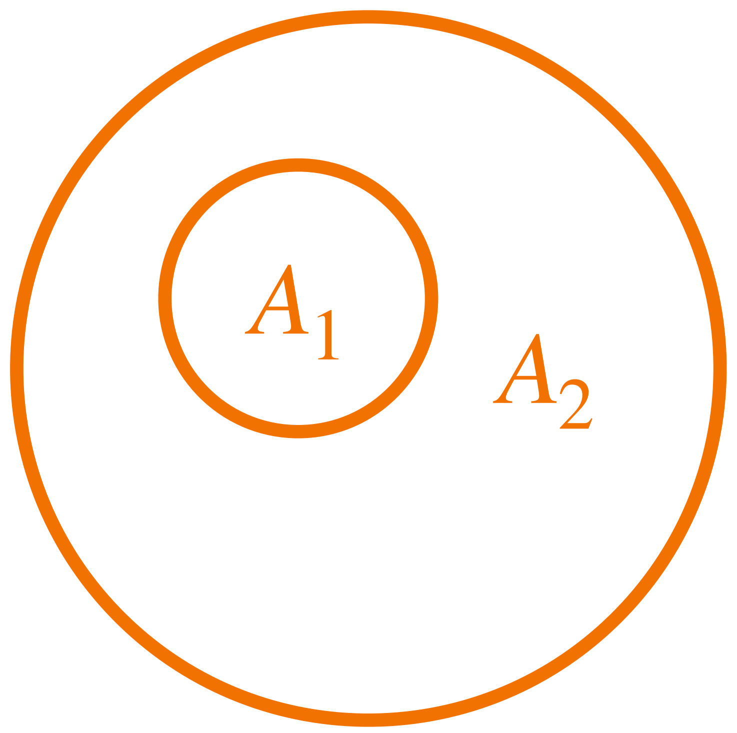
$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$

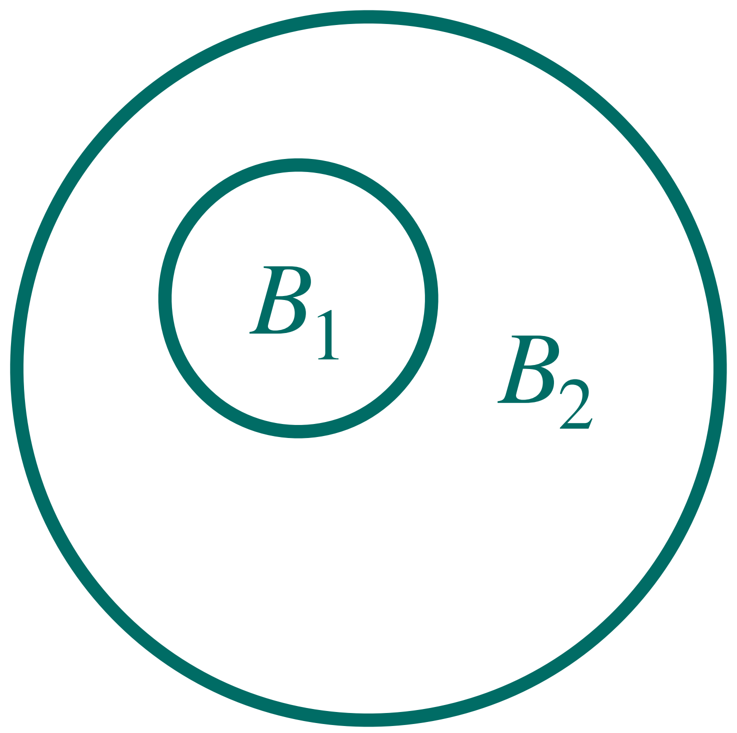


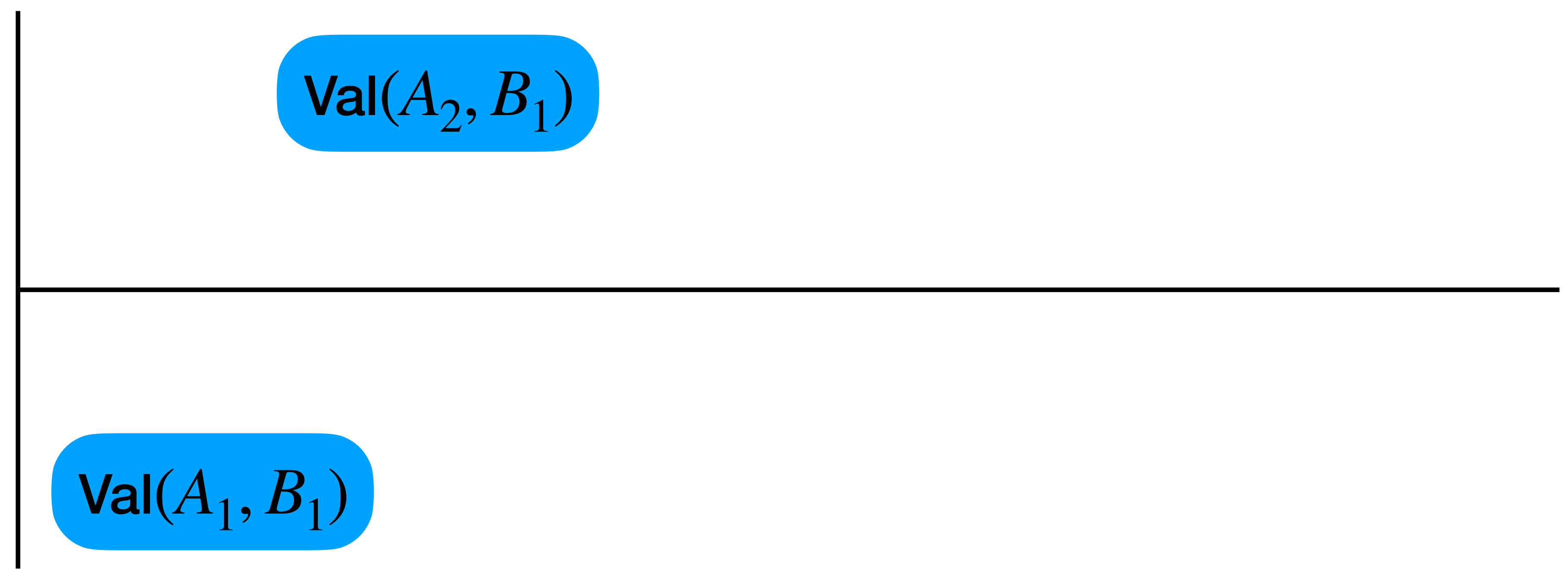
Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite Lit$(u)$

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$



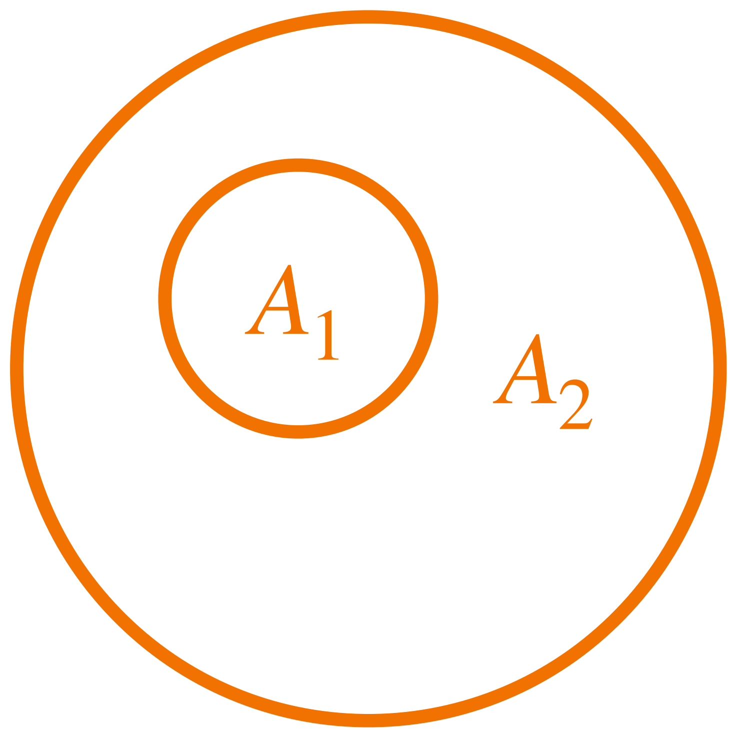


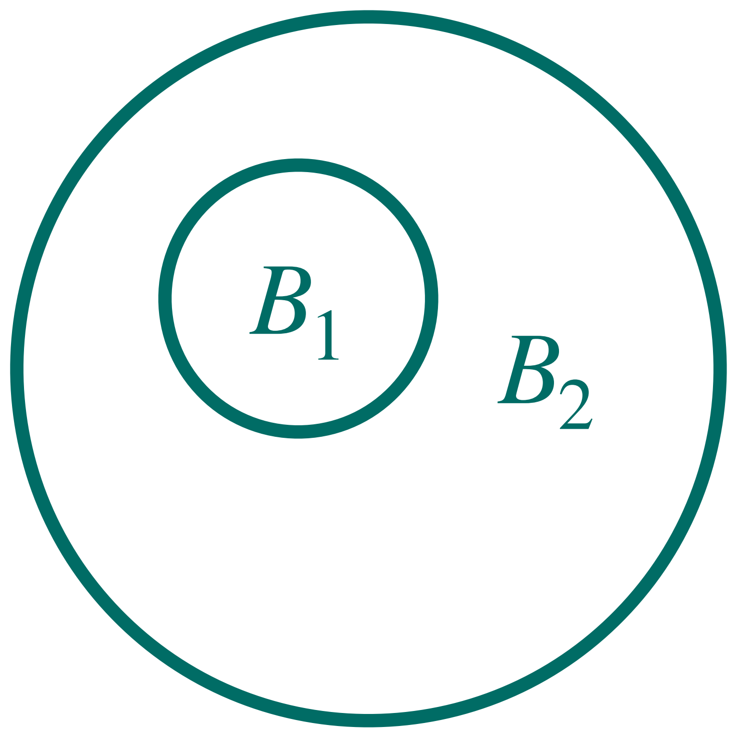


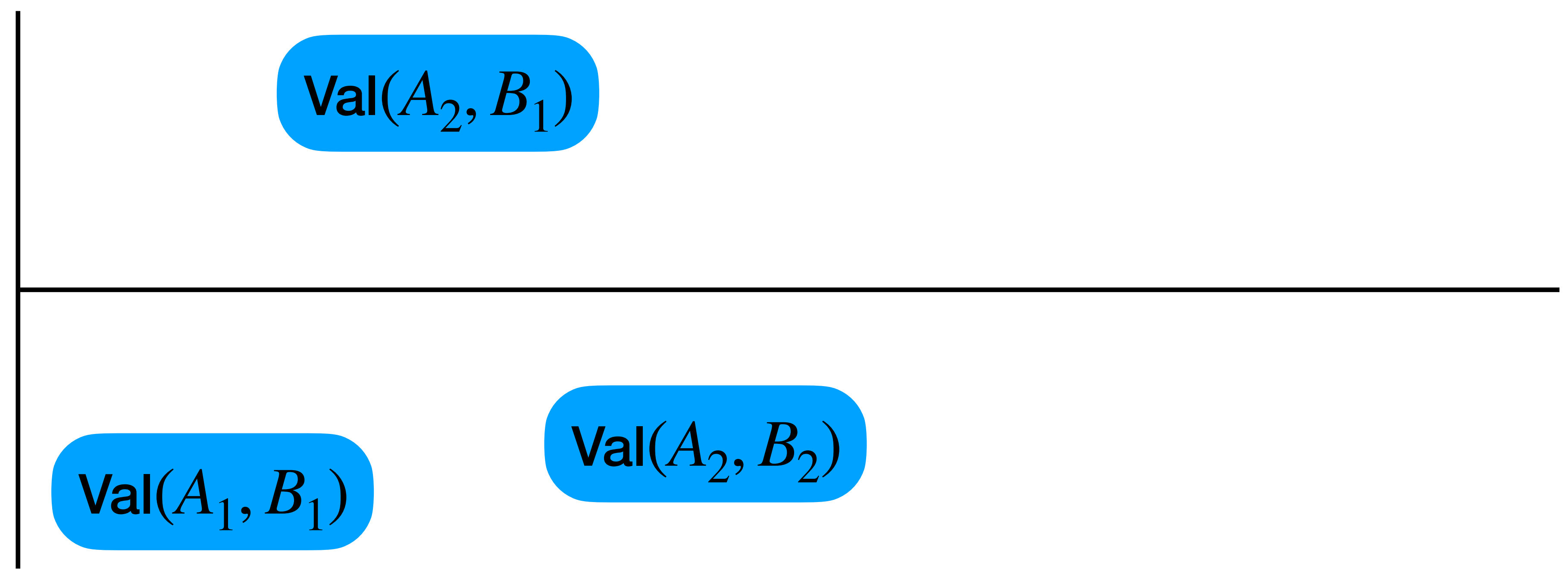
Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite $\mathrm{Lit}(u)$

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$





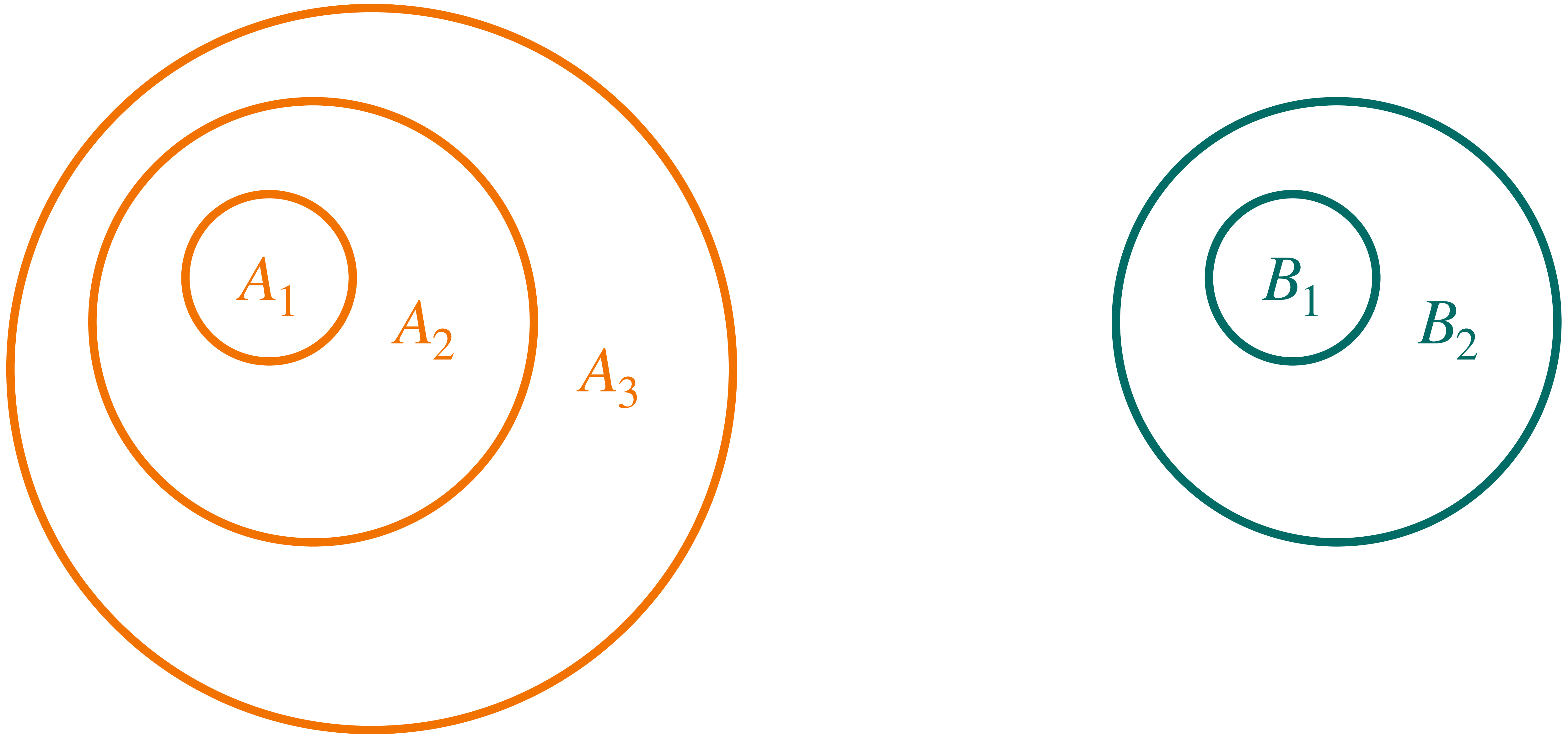


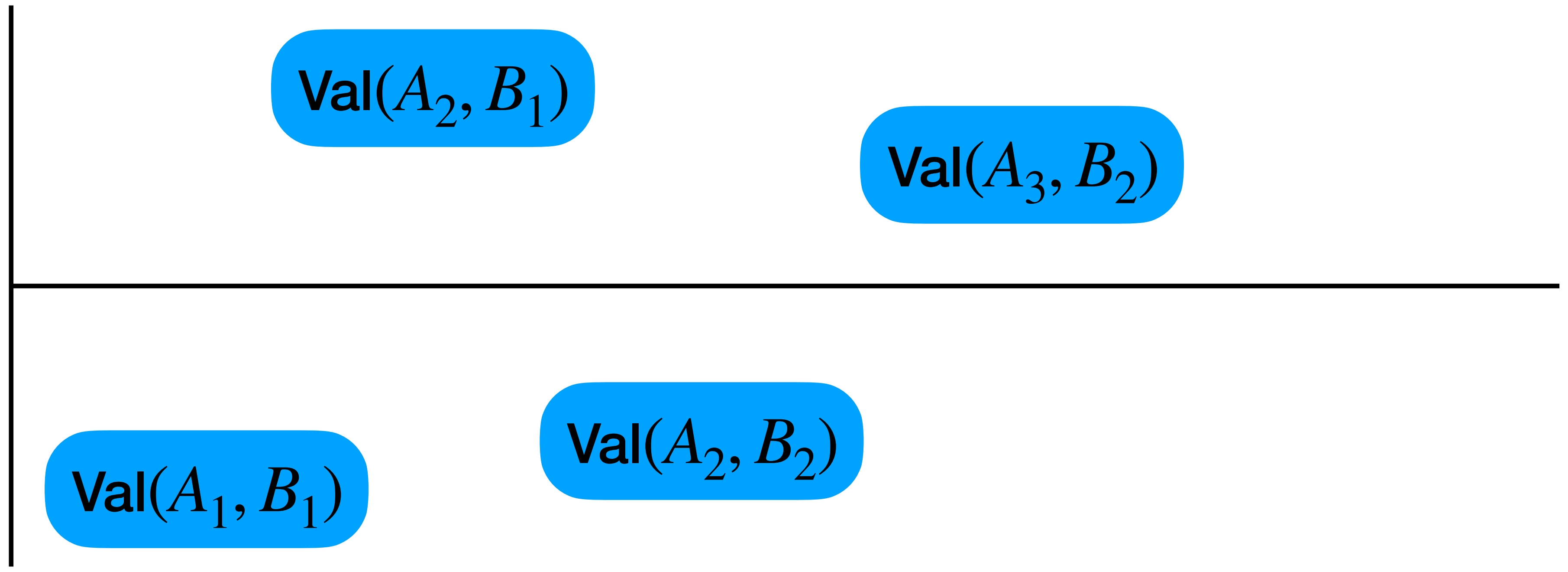
Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite Lit$(u)$

$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$

$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$

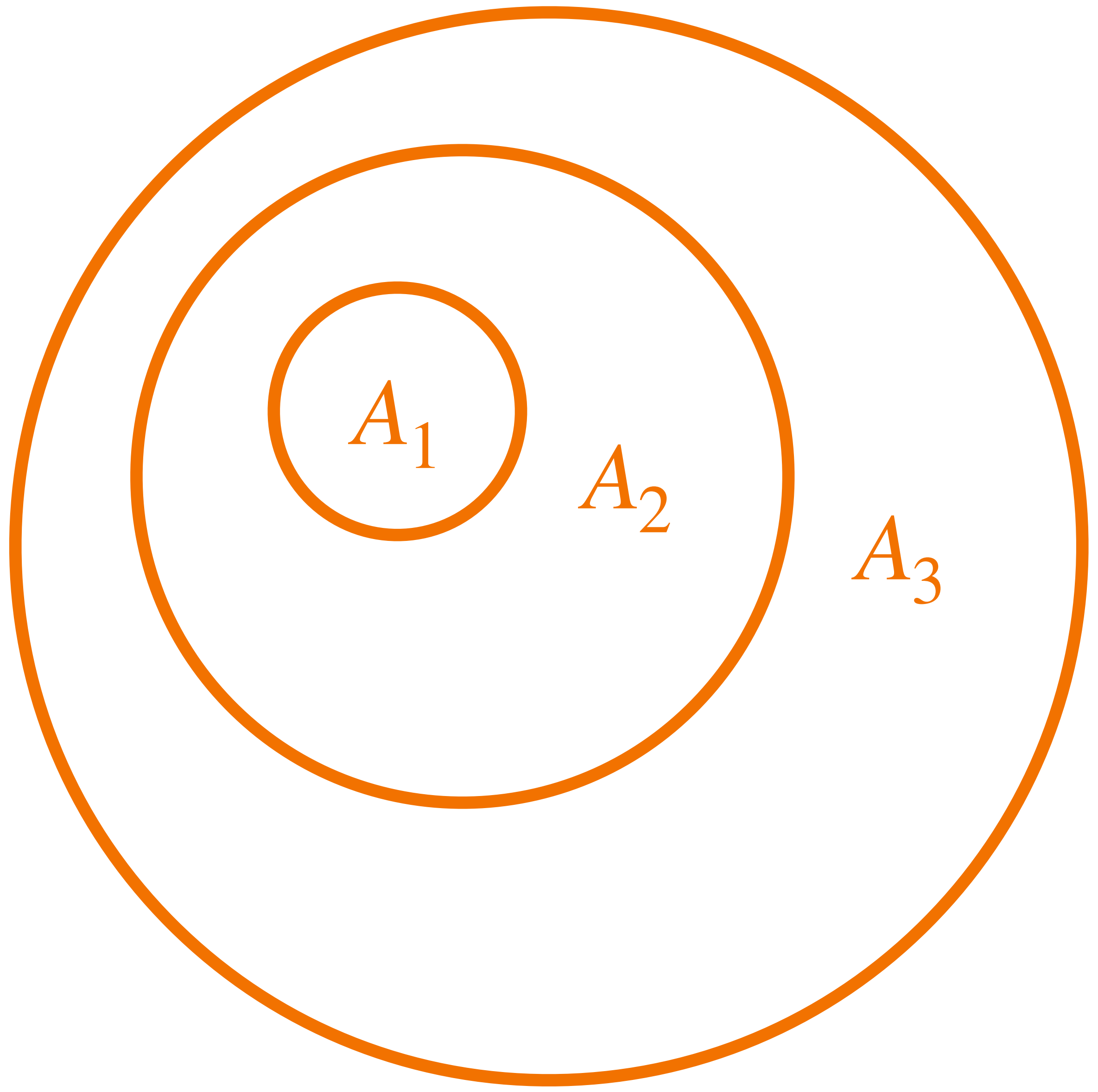


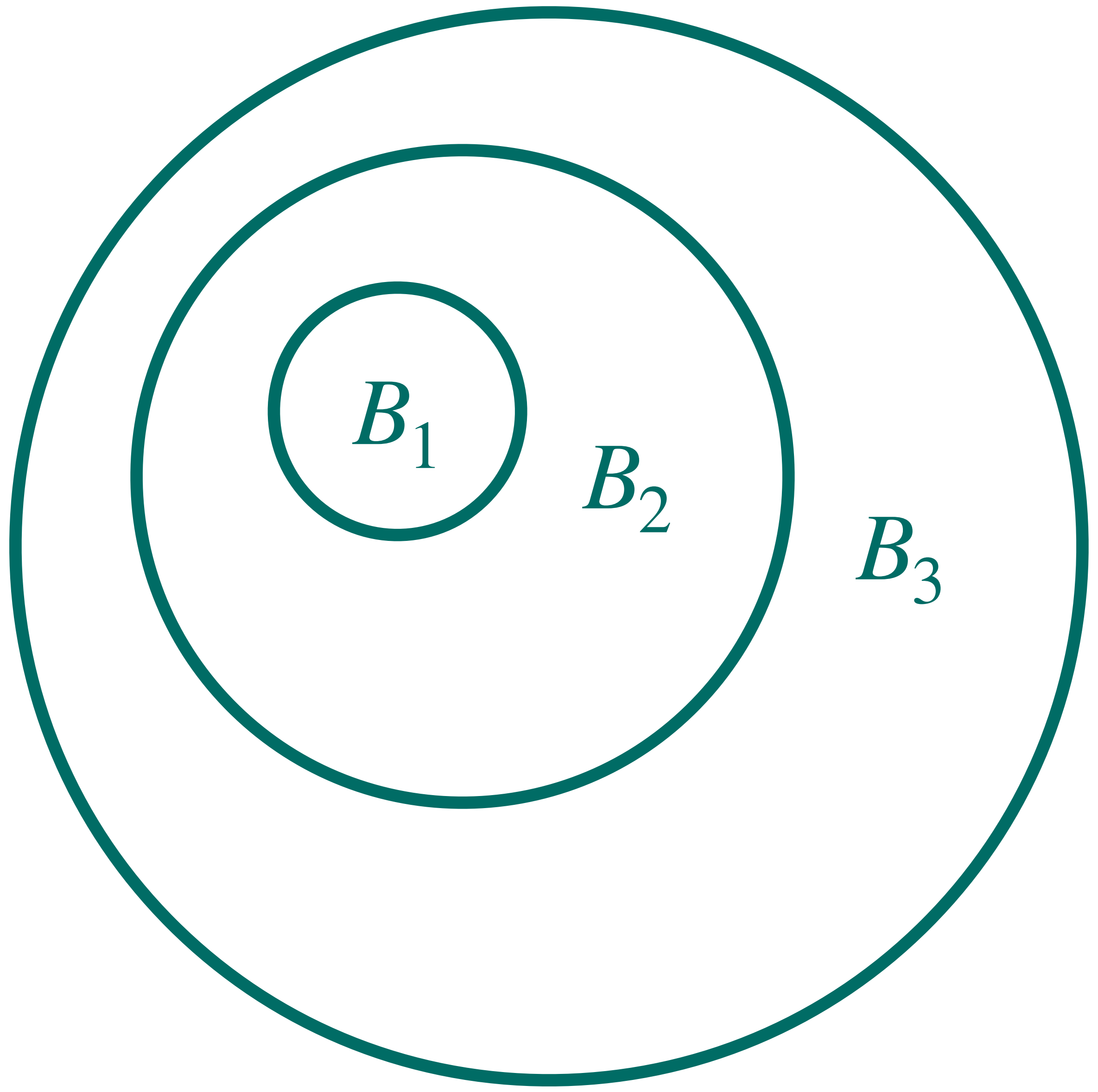


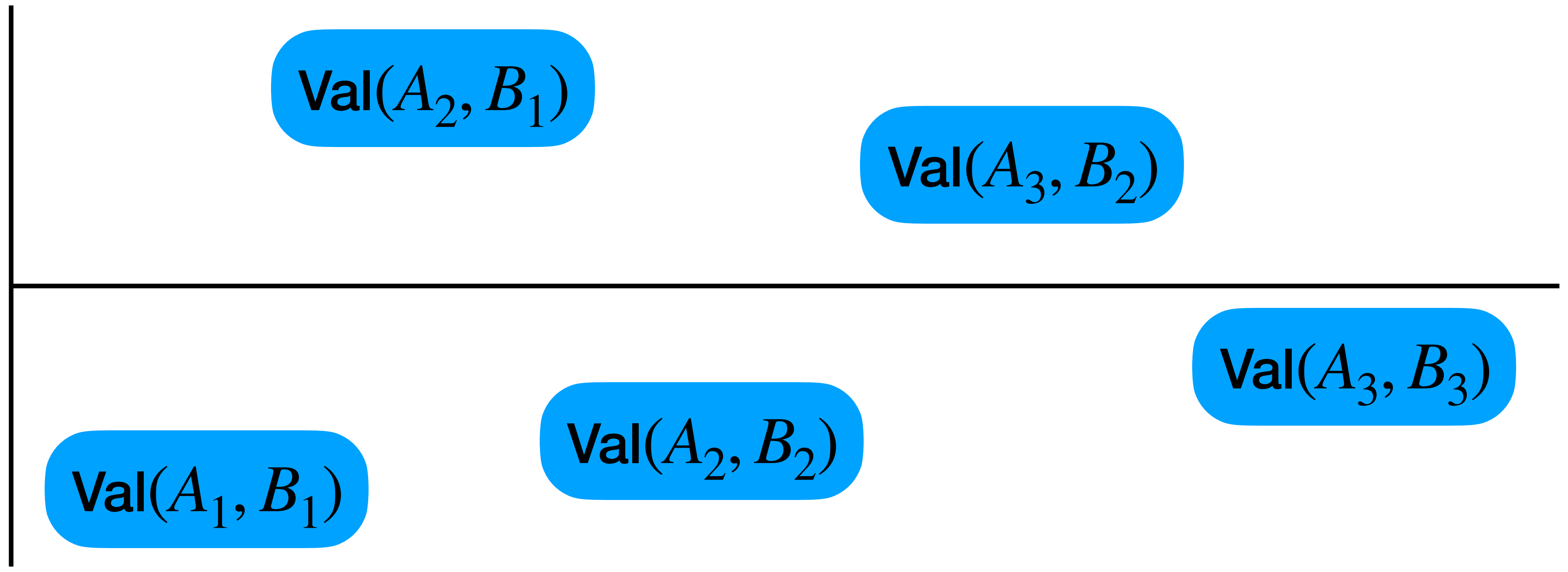
Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite Lit$(u)$

$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$
$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$



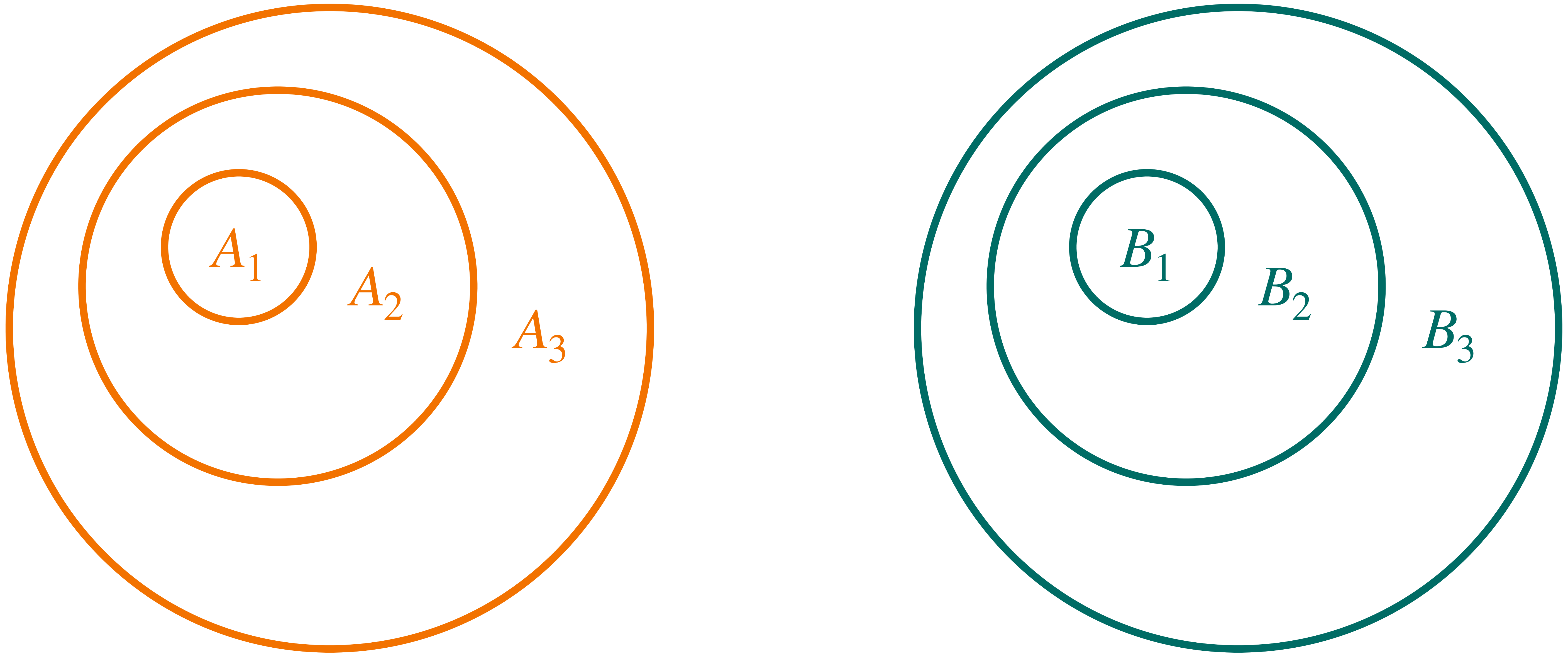


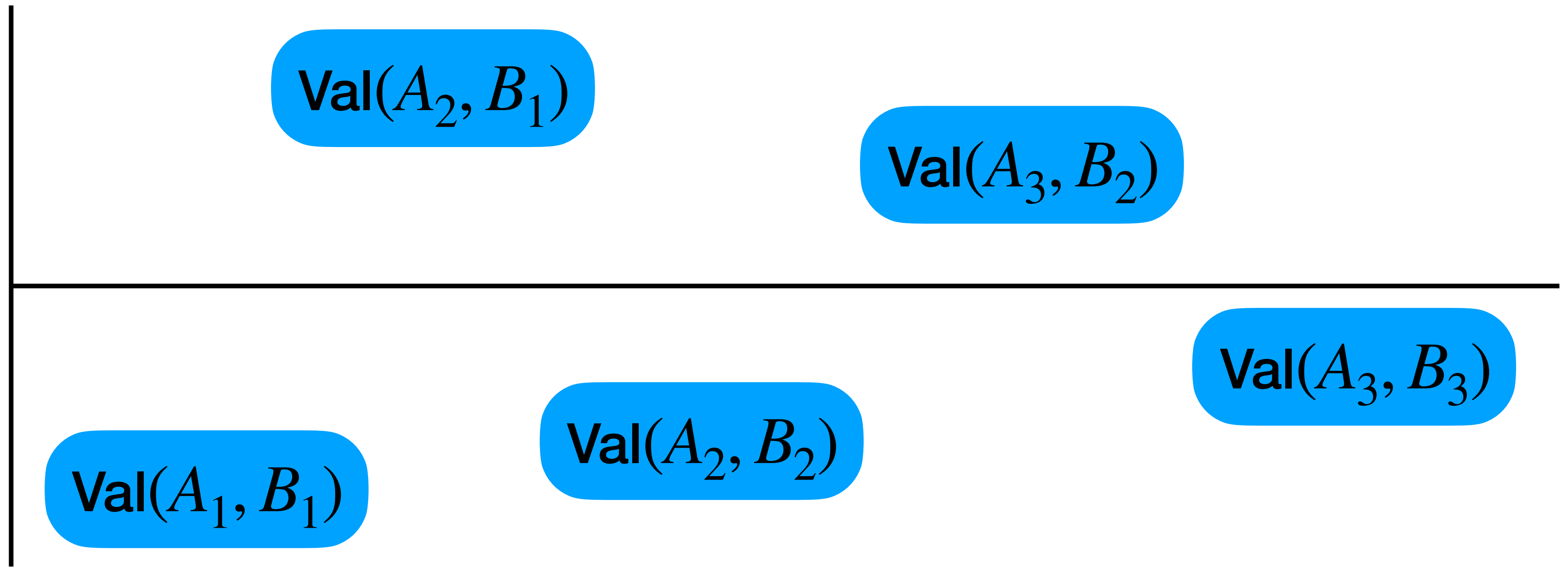


Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$, finite Lit$(u)$

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$



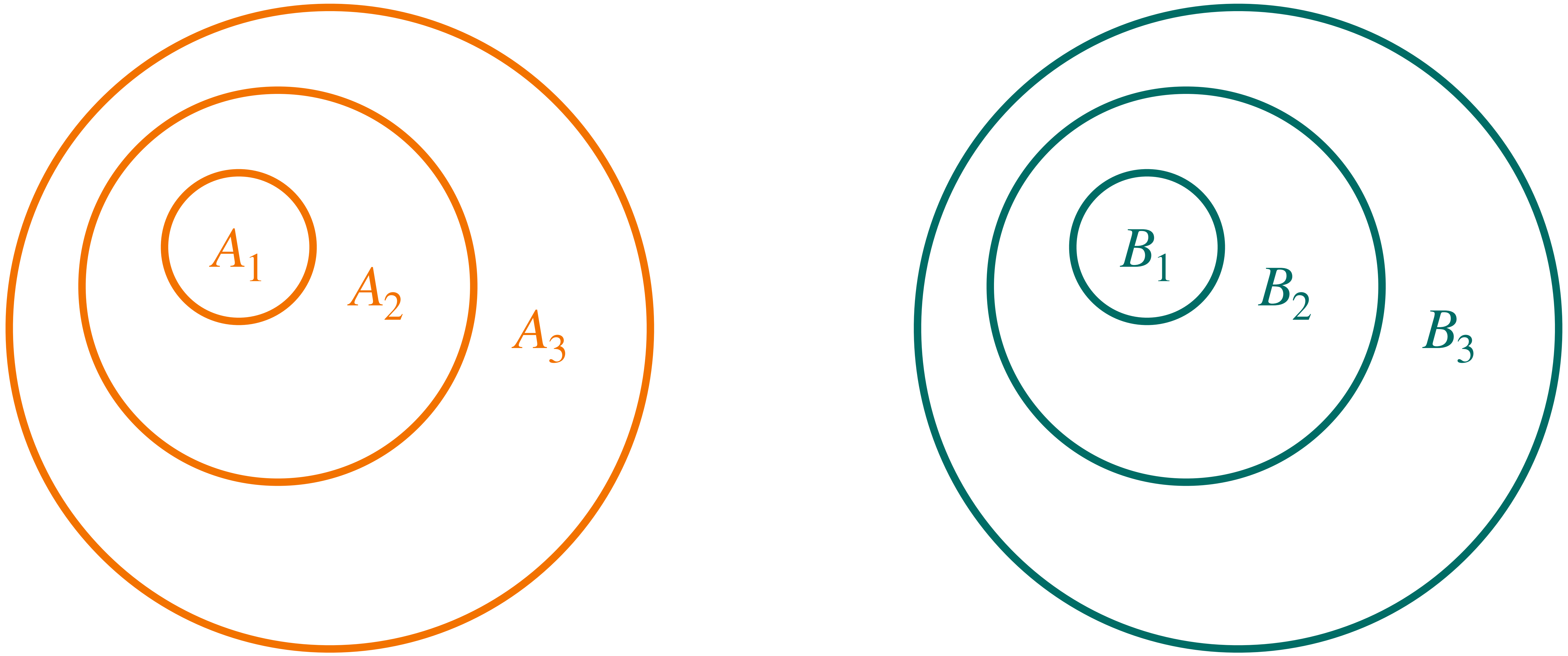


Proof idea
Nobody can increase value: equilibrium
Too many stages: construct large Littlestone tree

Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$ with finite Littlestone Dimension

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$

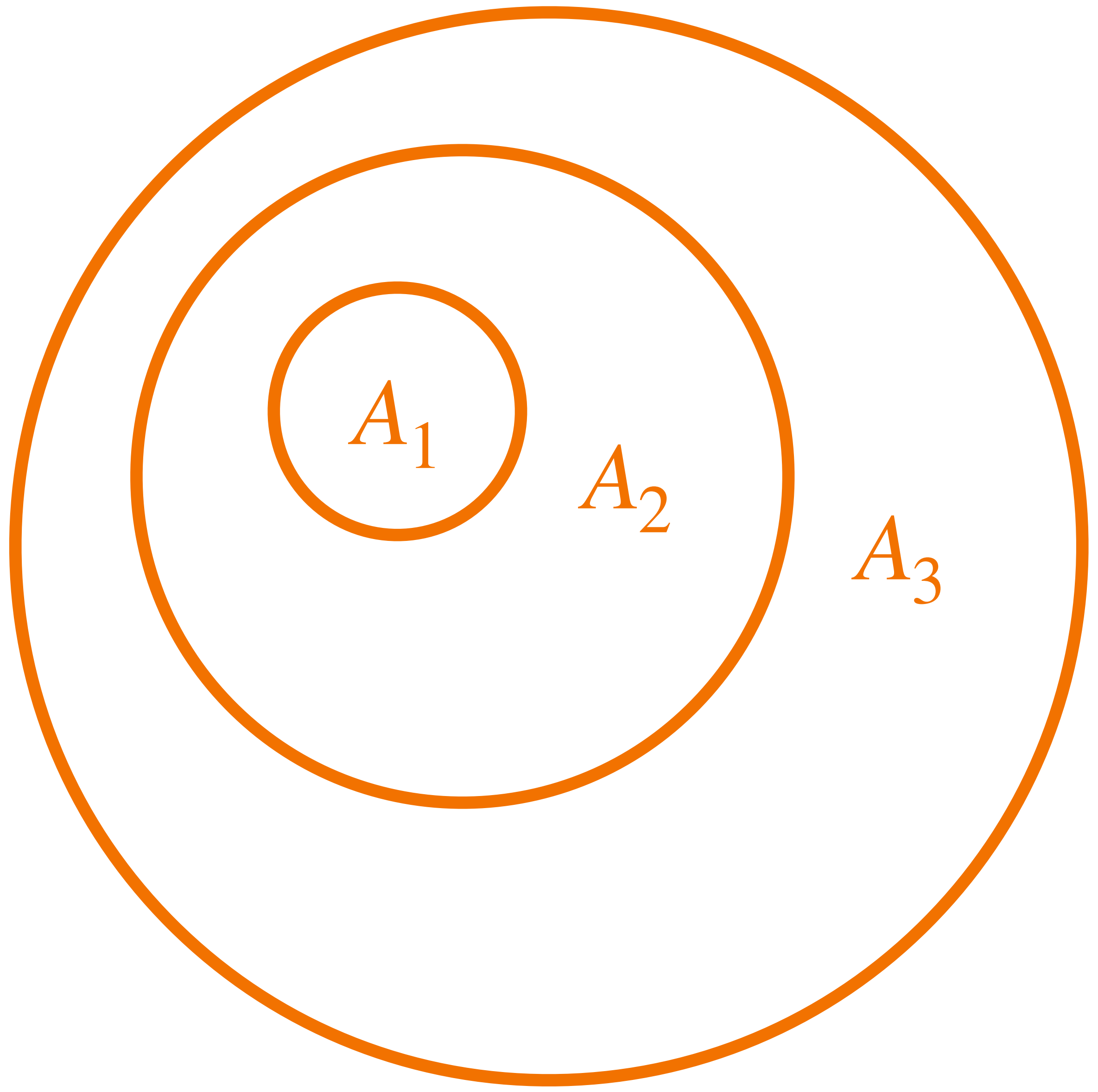


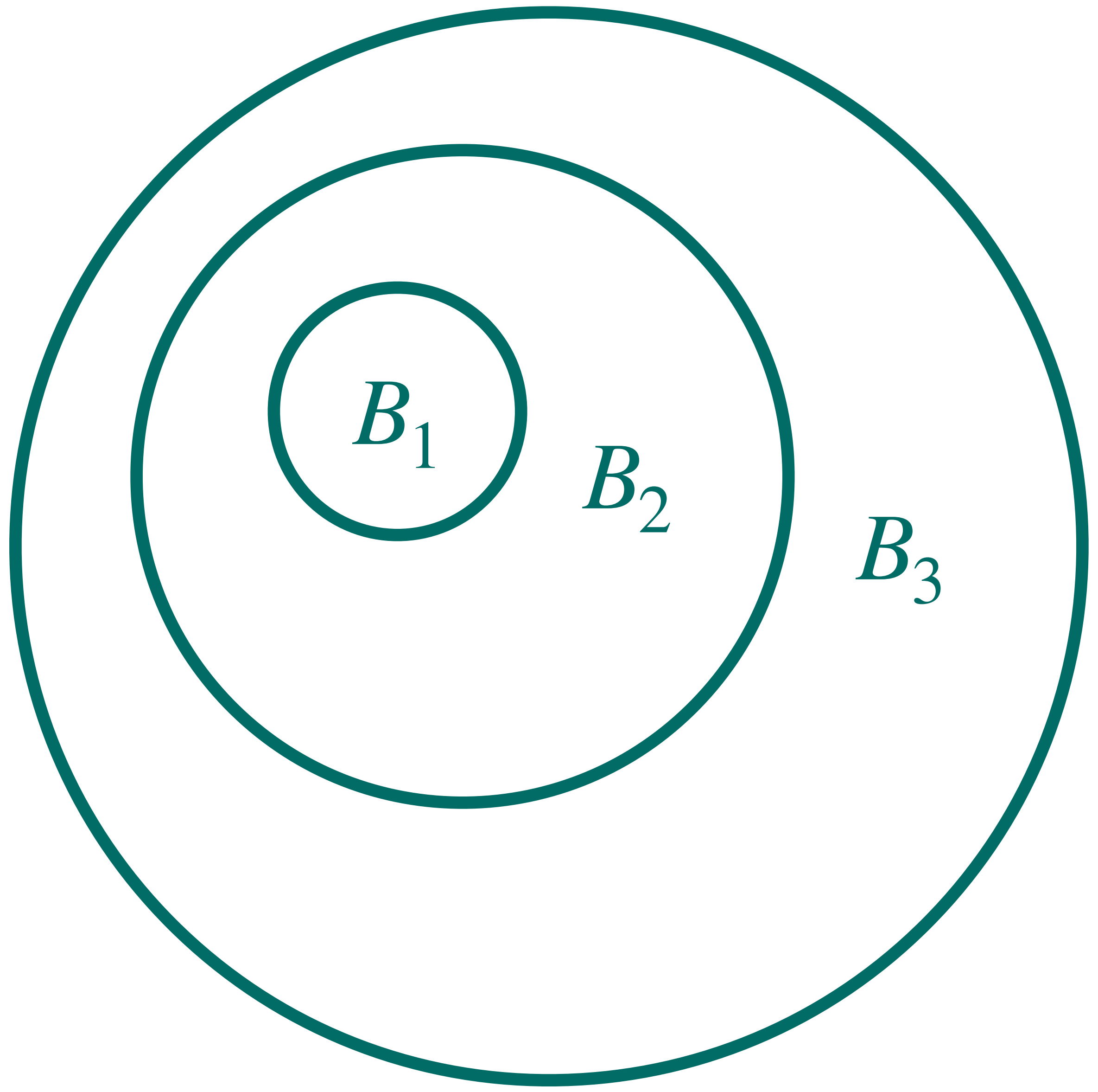
Double Oracle Algorithm:
Blotto [AHKK20]
MARL [DeepMind]

Algorithm (Two Player Zero Sum Games)

Game $u : \mathscr{A} \times \mathscr{B} \to \{0,1\}$ with finite Littlestone Dimension

$$A_1 \subseteq A_2 \subseteq \cdots \subset \mathscr{A}$$
$$B_1 \subseteq B_2 \subseteq \cdots \subset \mathscr{B}$$





Double Oracle Algorithm:
Blotto [AHKK20]
MARL [DeepMind]

	0-1 valued games	real valued games
2-player zero-sum ϵ-minmax	$C^{Lit(\mathscr{H})/\epsilon^2}$	$C^{sfat(\mathscr{H},\epsilon)/\epsilon^2}$
k-player general-sum ϵ-CCE	$C^{k \cdot Lit(\mathscr{H})/\epsilon^3}$	$C^{k \cdot sfat(\mathscr{H},\epsilon)/\epsilon^3}$

Questions?

maxfish@mit.edu
+1 917-434-7425

- [Littlestone 88] Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm.
- [Manurangsi 22] https://arxiv.org/pdf/2211.01443.pdf
- [Hanneke, Livni, Moran 21] https://arxiv.org/abs/2102.01646
- [AHKK20] https://arxiv.org/abs/2009.12185
- [DeepMind] http://mlanctot.info/files/papers/Lanctot_MARL_RLSS2019_Lille.pdf
- [FR98] https://www.sciencedirect.com/science/article/pii/S0890540198927092