MS&E 233
Game Theory, Data Science and AI
Lecture 6

Vasilis Syrgkanis

Assistant Professor

Management Science and Engineering

(by courtesy) Computer Science and Electrical Engineering

Institute for Computational and Mathematical Engineering

Computational Game Theory for Complex Games

1
- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2
- Basics of extensive-form games
- **Solving extensive-form games via online learning (T)**
- *HW3: implement agents to solve very simple variants of poker*

3
- General games and equilibria (T)
- Online learning in general games, multi-agent RL (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4
- Basics and applications of auction theory (T+A)
- Learning to bid in auctions via online learning (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5
- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: calculate equilibria in simple auctions, implement simple and optimal auctions, analyze revenue empirically*

6
- Optimizing mechanisms from samples (T)
- Online optimization of auctions and mechanisms (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples and in an online manner*

Further Topics

7
- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Solving Extensive Form Games via No-Regret Learning

Recap: No-Regret Learning in Sequence Form

- We have successfully turned imperfect information extensive form zero-sum games into a familiar object

$$\max_{\tilde{x} \in X} \min_{\tilde{y} \in Y} \tilde{x}^\top A \tilde{y}$$

- X, Y are convex sets, i.e., sequence-form strategies

- We can invoke minimax theorem to prove existence of equilibria
- We can calculate equilibria via LP duality
- We can calculate equilibria via no-regret learning!

Recap: Regret of FTRL

(FTRL) $$x_t = \operatorname*{argmax}_{x \in X} \boxed{\sum_{\tau < t} \langle x, u_\tau \rangle} - \boxed{\frac{1}{\eta} \mathcal{R}(x)}$$

1-strongly convex function of x that stabilizes the maximizer

Historical performance of always choosing strategy x

Theorem. Assuming the utility function at each period

$$f_t(x) = \langle x, u_t \rangle$$

is L-Lipschitz with respect to some norm $\|\cdot\|$ and the regularizer is 1-strongly convex with respect to the same norm then

$$\text{Regret} - \text{FTRL}(T) \leq \boxed{\eta L} + \boxed{\frac{1}{\eta T} \left(\max_{x \in X} \mathcal{R}(x) - \min_{x \in X} \mathcal{R}(x) \right)}$$

Average stability induced by regularizer

Average loss distortion caused by regularizer

Recap: Regularizer for the Treeplex Space X

$$U_{t-1} = \sum_{\tau<t} u_\tau$$

- The only thing we are missing is a good Regularizer for X

- **Desiderata.** Be strongly convex in x within X and for the optimization problem to be fast to solve

$$\tilde{x}_t = \operatorname*{argmax}_{\tilde{x}\in X} \sum_{\tau<t} \langle \tilde{x}, u_\tau \rangle - \frac{1}{\eta}\mathcal{R}(\tilde{x}) = \operatorname*{argmax}_{\tilde{x}\in X} \langle \tilde{x}, \boxed{U_{t-1}} \rangle - \frac{1}{\eta}\mathcal{R}(\tilde{x})$$

- X is no longer a "simplex", so entropy is not a good Regularizer

Dilated Entropy

- X is a combination of *scaled simplices*, i.e., $\tilde{x} = \left(\tilde{x}^j\right)_{j \in \mathcal{J}_1}$
- $\tilde{x}^j = (\tilde{x}_a)_{a \in A_j}$: sequence-form strategies for actions in infoset $j \in \mathcal{J}_1$

$$\tilde{x}^j \in \tilde{x}_{p_j} \cdot \Delta_j \quad \Leftrightarrow \quad \tilde{x}^j / \tilde{x}_{p_j} \in \Delta_j$$

- Consider a *weighted combination of local negative entropies*

$$\mathcal{R}(\tilde{x}) := \sum_j \beta_j \, \tilde{x}_{p_j} \, \mathrm{H}\left(\tilde{x}^j / \tilde{x}_{p_j}\right), \qquad \mathrm{H}(u) = \sum_i u_i \log(u_i)$$

Lies in a simplex Δ_j

Equivalent to the behavioral strategy x^j

Negative Entropy

- $\mathcal{R}(\tilde{x})$ is $1/M$ strongly convex w.r.t. ℓ_1 norm, where $M = \max_{\tilde{x} \in X} \|\tilde{x}\|_1$, for appropriate choice of β_j based on game tree structure

Solving the Optimization Problem

- Optimization problem decomposes into local simplex problems

$$\sum_{j \in \mathcal{J}_1} \left\langle \tilde{x}^j, U_{t-1}^j \right\rangle - \underbrace{\boxed{\frac{1}{\eta}\beta_j}}_{:= \frac{1}{\eta_j}} \tilde{x}_{p_j} \, \mathrm{H}\left(\frac{\tilde{x}^j}{\tilde{x}_{p_j}}\right) = \sum_{j \in \mathcal{J}_1} \tilde{x}_{p_j} \left\{ \left\langle \frac{\tilde{x}^j}{\tilde{x}_{p_j}}, U_{t-1}^j \right\rangle - \frac{1}{\eta_j} \mathrm{H}\left(\frac{\tilde{x}^j}{\tilde{x}_{p_j}}\right) \right\}$$

- Quantity $\frac{\tilde{x}^j}{\tilde{x}_{p_j}}$ is essentially the behavioral strategy x^j at infoset j

$$\sum_{j \in \mathcal{J}_1} \tilde{x}_{p_j} \left\{ \left\langle x^j, U_{t-1}^j \right\rangle - \frac{1}{\eta_j} \mathrm{H}\left(x^j\right) \right\}$$

- Quantity x^j over simplex Δ_j is independent of solution x_a for all ancestral actions and only appears in subsequent infosets

Solving the Optimization Problem

- Decomposes in local max over behavioral strategies x^j solved bottom up

$$V^j = \max_{x^j \in \Delta_j} \left\langle x^j, U^j_{t-1} \right\rangle - \frac{1}{\eta_j} H\left(x^j\right) \Rightarrow \begin{cases} x^j \propto \exp\left(\eta_j U^j_{t-1}\right) \\ V^j = \log \sum_{a \in A_j} \exp\left(\eta_j U^a_{t-1}\right) = \text{softmax}_{\eta_j}\left(U^j_{t-1}\right) \end{cases}$$

- Value V^j multiplies $\tilde{x}_{p_j}$; when solving for $\tilde{x}_{p_j}$ we need to take it into account. If $p_j \in A_k$

$$\max_{x^k \in \Delta_k} \left\langle \tilde{x}^k, U^k_{t-1} \right\rangle - \eta_k\, \tilde{x}_{p_k}\, \mathrm{H}\left(\frac{\tilde{x}^k}{\tilde{x}_{p_k}}\right) + \tilde{x}_{p_j} V^j + \cdots$$

- Add V^j to "cumulative utility" Q_{p_j} (initialized at U_{t-1,p_j}) associated with p_j

$$Q_{p_j} \leftarrow Q_{p_j} + V^j$$

Sum: Nash via FTRL with Dilated Entropy

Each player chooses $\tilde{x}_t, \tilde{y}_t$ based on FTRL with dilated entropy

- For x-player $u_t = A\tilde{y}_t$ and $U_t = U_{t-1} + u_t$ and initialize $Q = U_t$
- Traverse the tree bottom-up; for each infoset $j \in \mathcal{J}_1$
  $$x_{t+1}^j \propto \exp\left(\eta_j Q^j\right), \qquad V^j = \mathrm{softmax}_{\eta_j}\left(Q^j\right), \qquad Q_{p_j} \leftarrow Q_{p_j} + V^j$$
- Define sequence-form strategies top-down: $\tilde{x}_{t+1}^j = \tilde{x}_{p_j} \cdot x_{t+1}^j$

Similarly, for y player

Return average of sequence-form strategies as equilibrium

Interpreting utility vector

$$u_{t,a} = A\tilde{y}_t = \sum_{a' \in A_{P2}} A_{a,a'} \tilde{y}_{t,a'}$$

$A_{a,a'}$ is zero if the combination of a, a' does not lead to a leaf node

$$u_{t,a} = \sum_{\substack{\text{Leafs } z:\ a \text{ was last P1 action} \\ a' \text{ was last P2 action}}} u(z) \Pr\begin{pmatrix}\text{Chance chooses} \\ \text{sequence on} \\ \text{path to } z\end{pmatrix} \Pr\begin{pmatrix}\text{P2 plays} \\ \text{sequence} \\ \text{leading to } a'\end{pmatrix}$$

Interpretation. If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then don't make any other moves*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(\qquad\qquad\qquad\qquad \right)$$

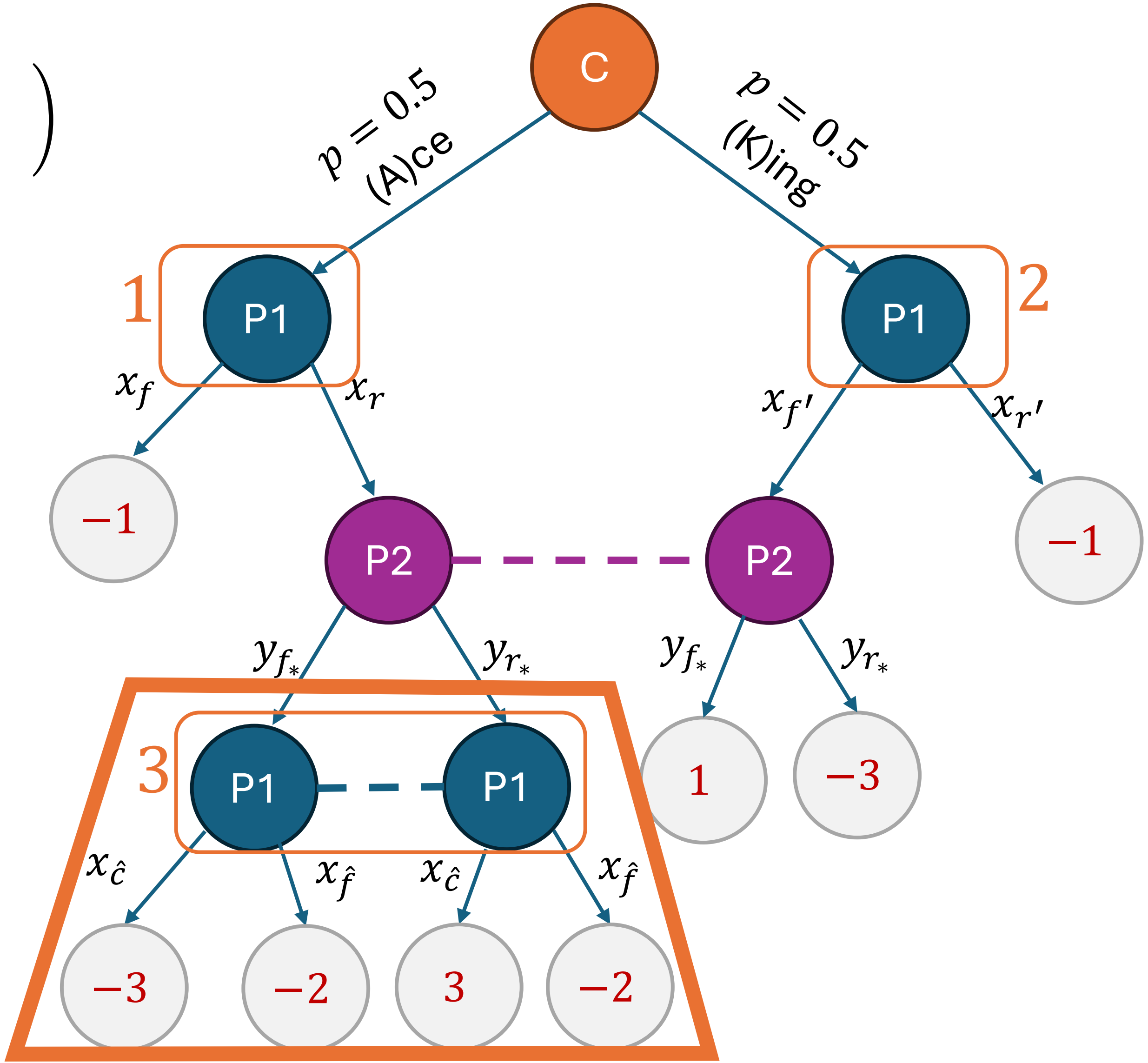


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= (u_{\hat{c}}, u_{\hat{f}}) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

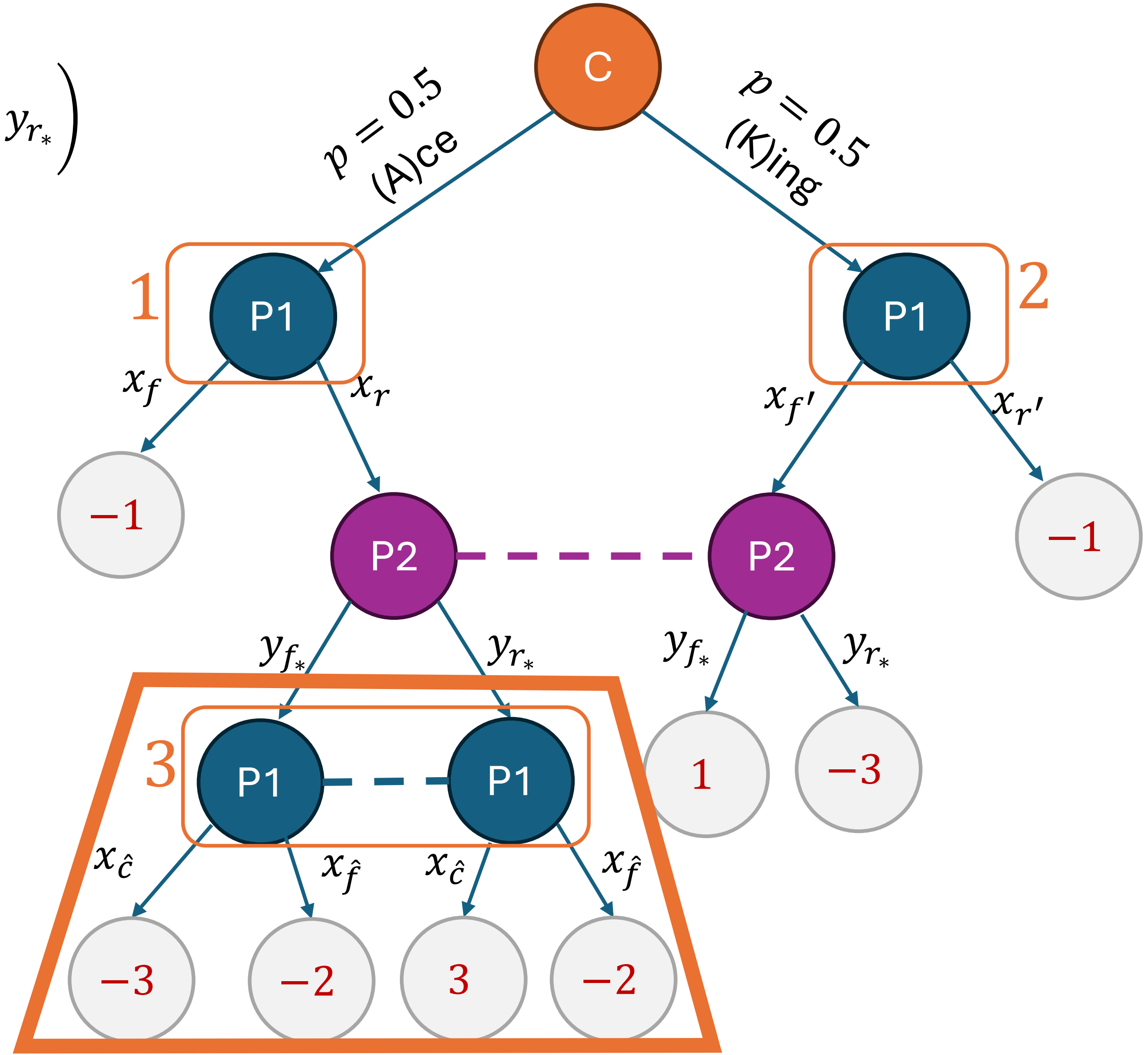


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, Q^3\right)$$

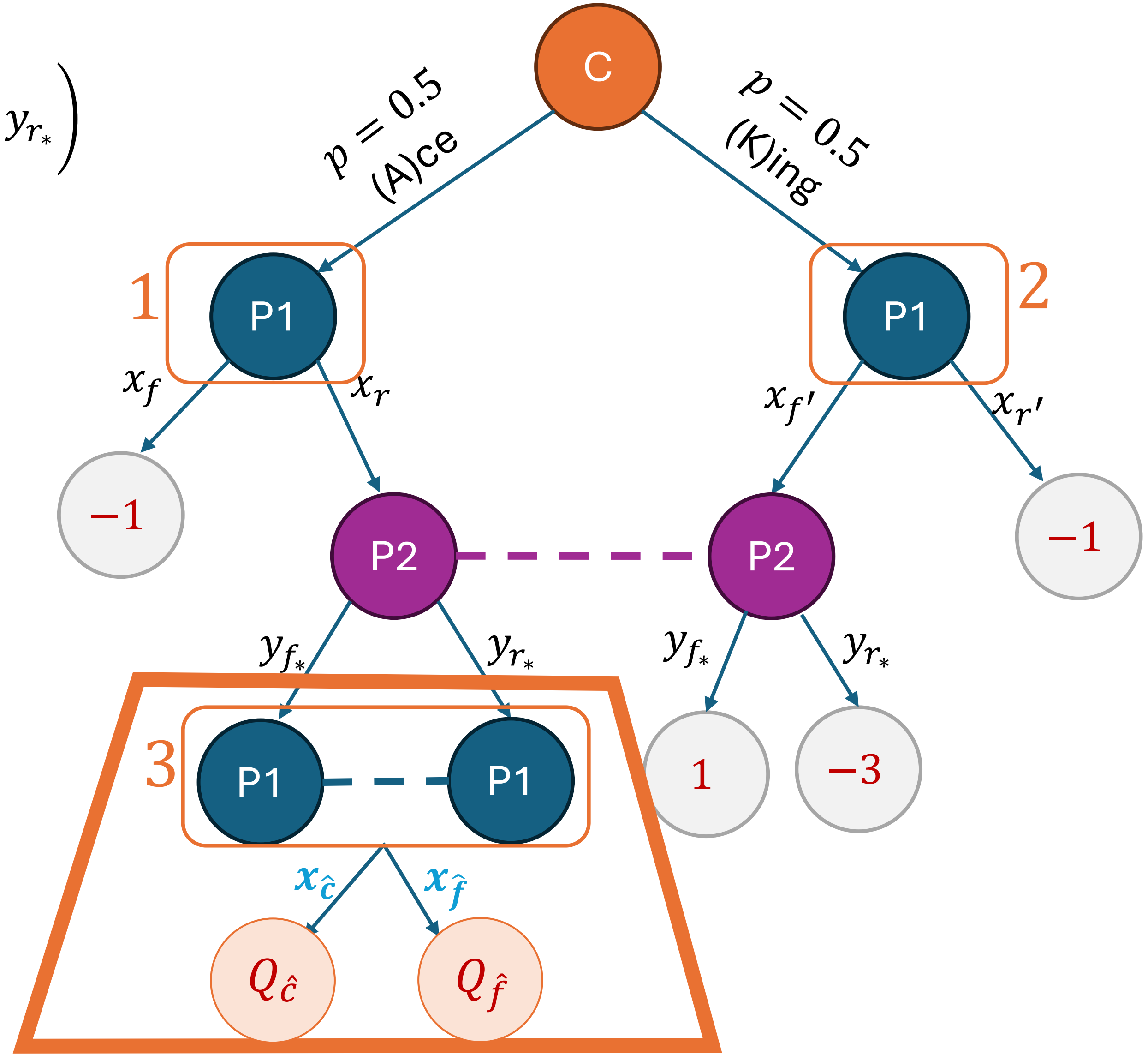


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, Q^3\right)$$

$$V^3 = \text{softmax}(\eta_3 Q^3)$$

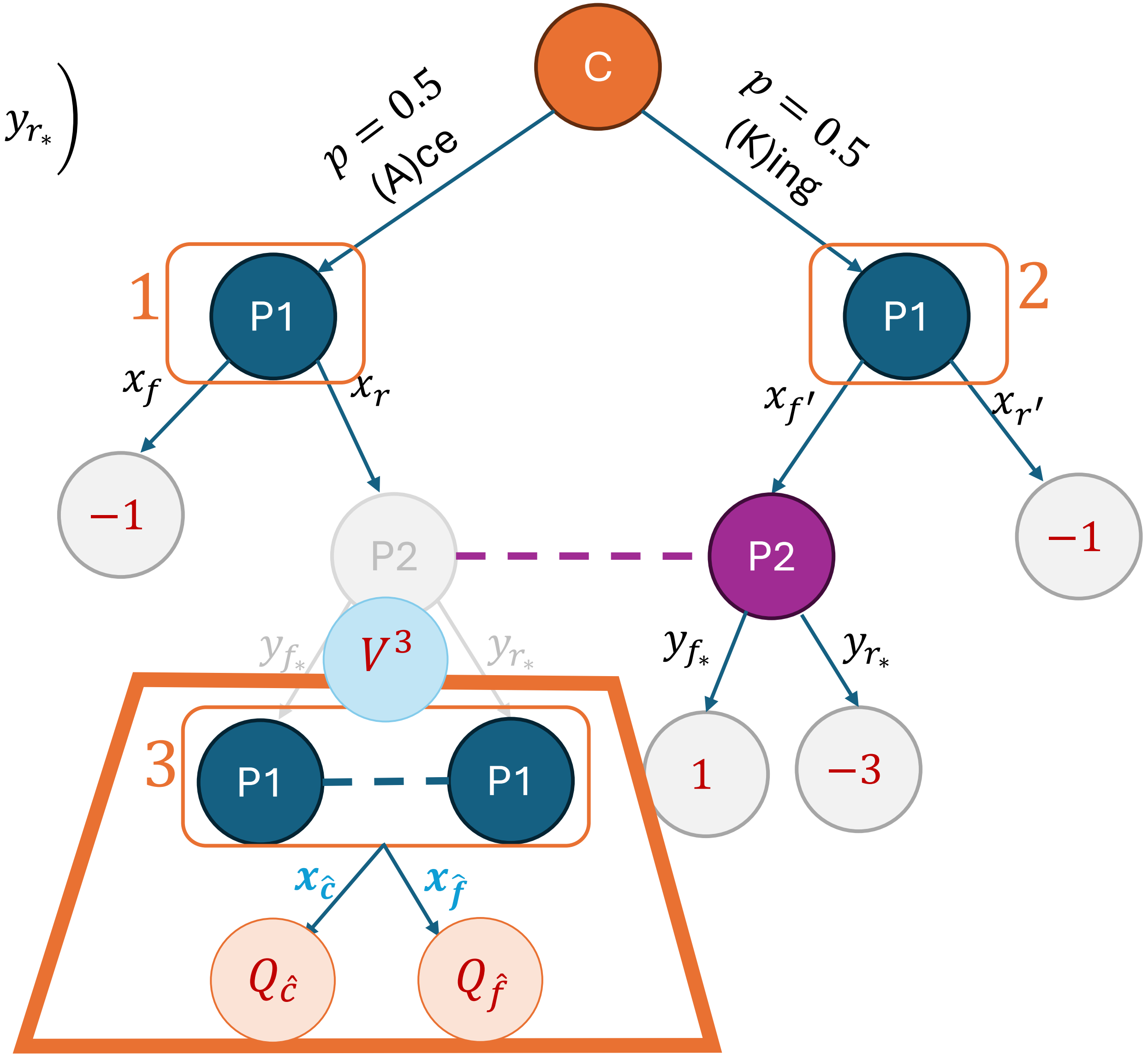


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, Q^3\right)$$

$$V^3 = \mathrm{softmax}(\eta_3 Q^3)$$

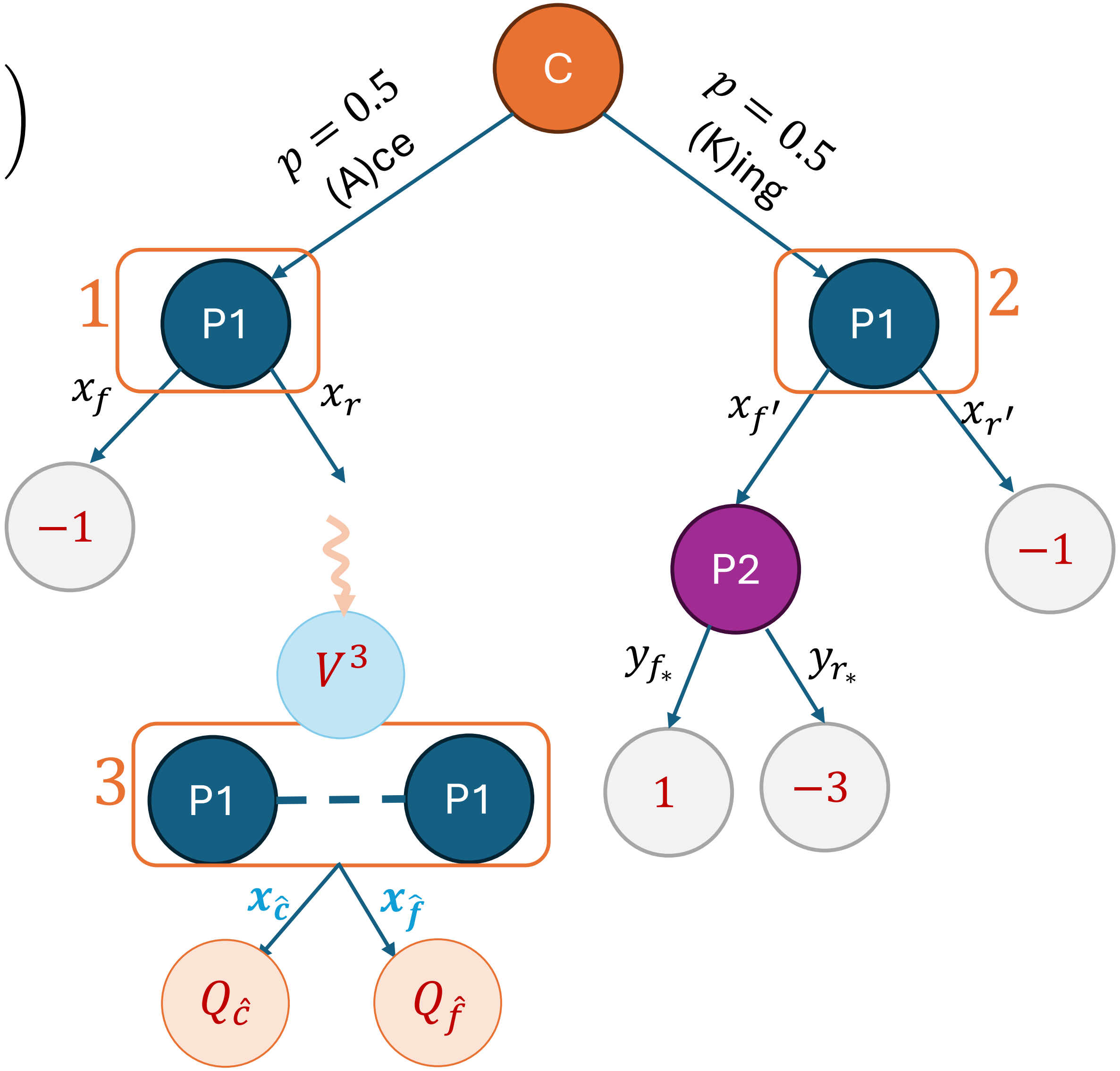


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, Q^3\right)$$

$$V^3 = \mathrm{softmax}(\eta_3 Q^3)$$

- Go to **Infoset 1**

$$U^1 \mathrel{+}= (u_f, u_r) = \left(\qquad \right)$$

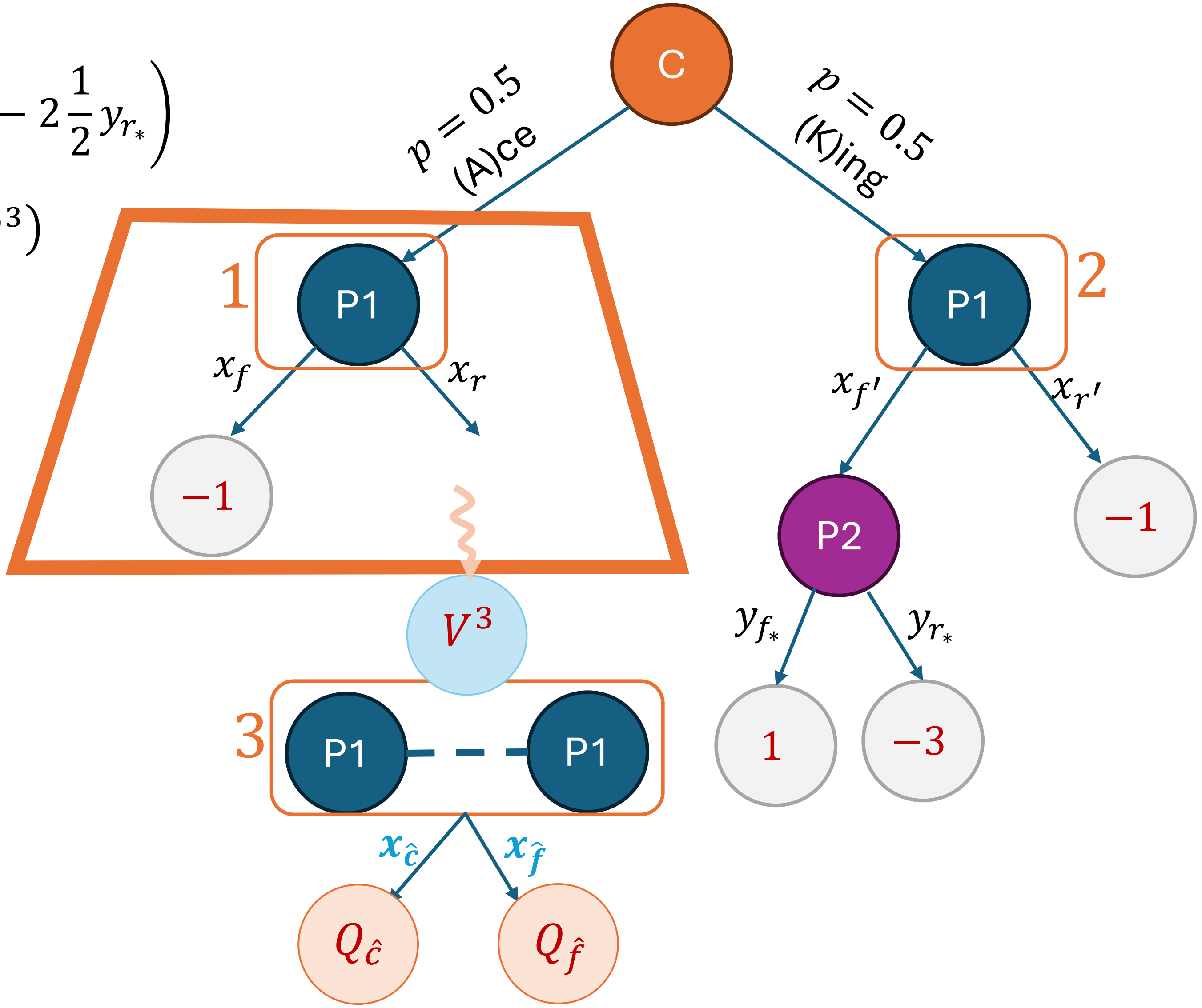


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, Q^3\right)$$

$$V^3 = \mathrm{softmax}(\eta_3 Q^3)$$

- Go to **Infoset 1**

$$U^1 \mathrel{+}= \left(u_f, u_r\right) = \left(-1\frac{1}{2}, 0\right)$$

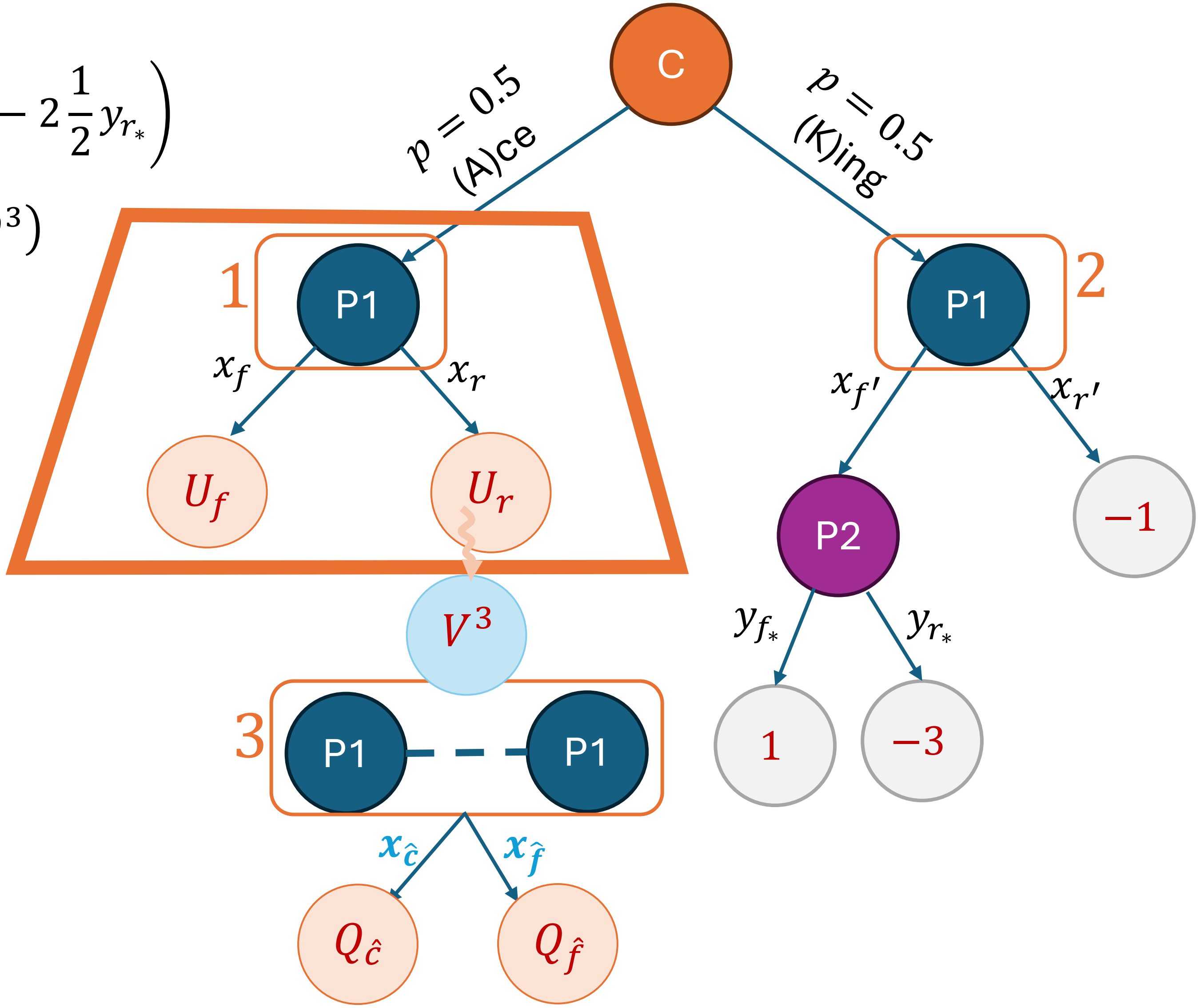


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp(\eta_3\, Q^3)$$

$$V^3 = \mathrm{softmax}(\eta_3 Q^3)$$

- Go to **Infoset 1**

$$U^1 \mathrel{+}= (u_f, u_r) = \left(-1\frac{1}{2}, 0\right)$$

$$Q^1 = U^1 + (0, V^3) = \left(-1\frac{1}{2}, V^3\right)$$

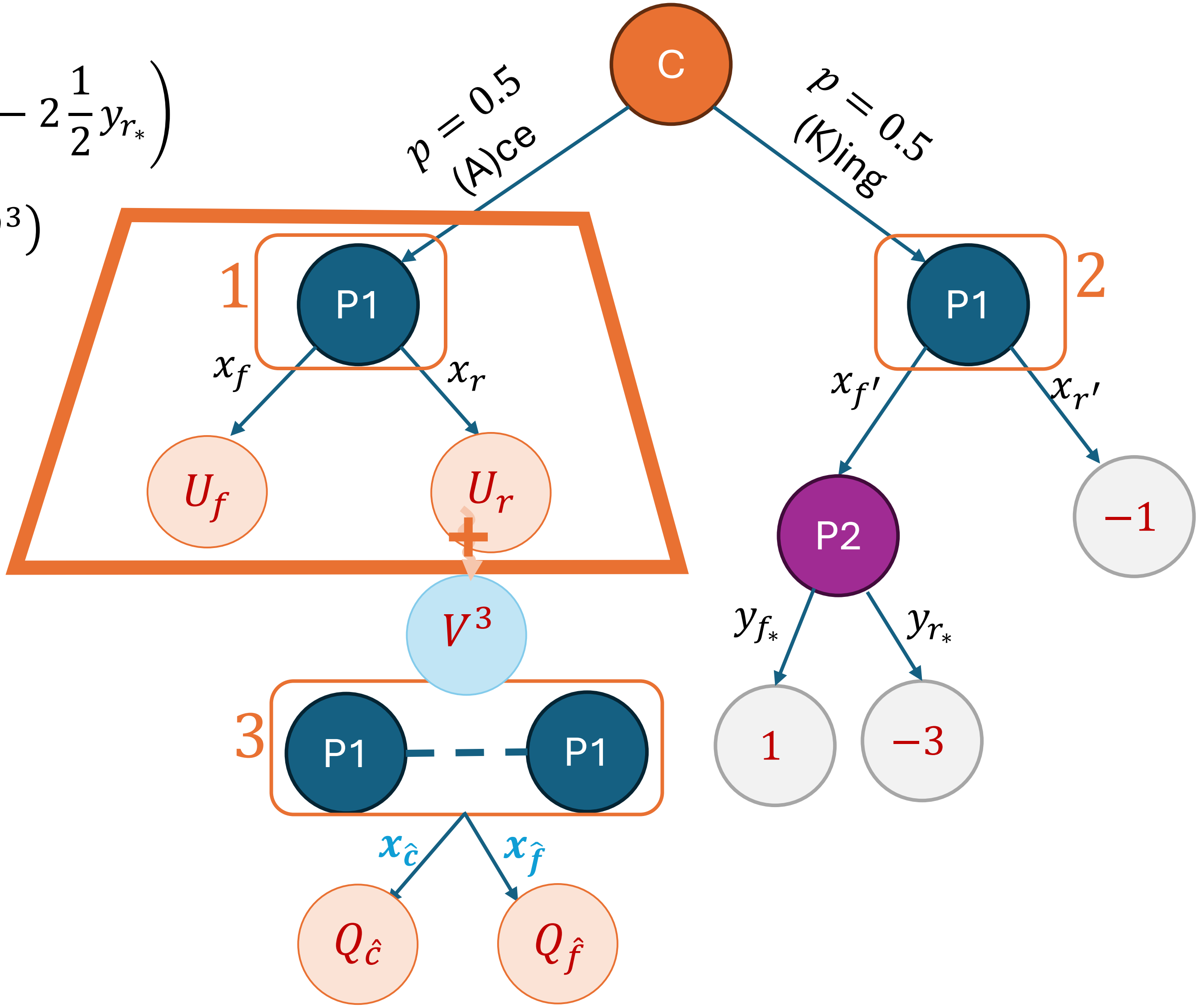


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, Q^3\right)$$

$$V^3 = \text{softmax}(\eta_3 Q^3)$$

- Go to **Infoset 1**

$$U^1 \mathrel{+}= \left(u_f, u_r\right) = \left(-1\frac{1}{2}, 0\right)$$

$$Q^1 = U^1 + \left(0, V^3\right) = \left(-1\frac{1}{2}, V^3\right)$$

$$x^1 = \left(x_f, x_r\right) \propto \exp\left(\eta_1\, Q^1\right)$$

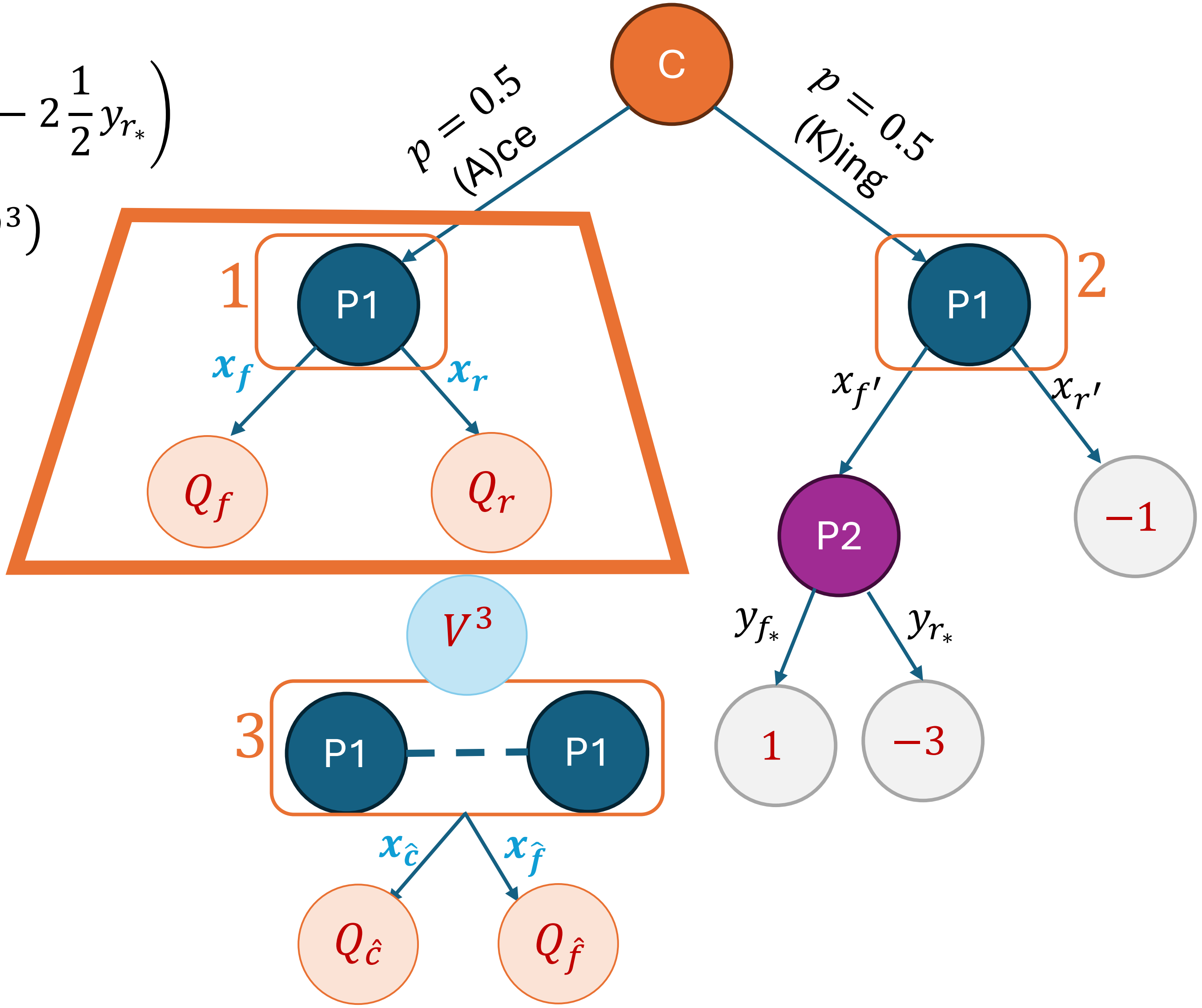


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= (u_{\hat{c}}, u_{\hat{f}}) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = (x_{\hat{c}}, x_{\hat{f}}) \propto \exp(\eta_3\, Q^3)$$

$$V^3 = \text{softmax}(\eta_3 Q^3)$$

- Go to **Infoset 1**

$$U^1 \mathrel{+}= (u_f, u_r) = \left(-1\frac{1}{2}, 0\right)$$

$$Q^1 = U^1 + (0, V^3) = \left(-1\frac{1}{2}, V^3\right)$$

$$x^1 = (x_f, x_r) \propto \exp(\eta_1\, Q^1)$$

- Go to **Infoset 2**

$$U^2 \mathrel{+}= (u_{f'}, u_{r'}) = \left(\qquad\qquad\qquad \right)$$

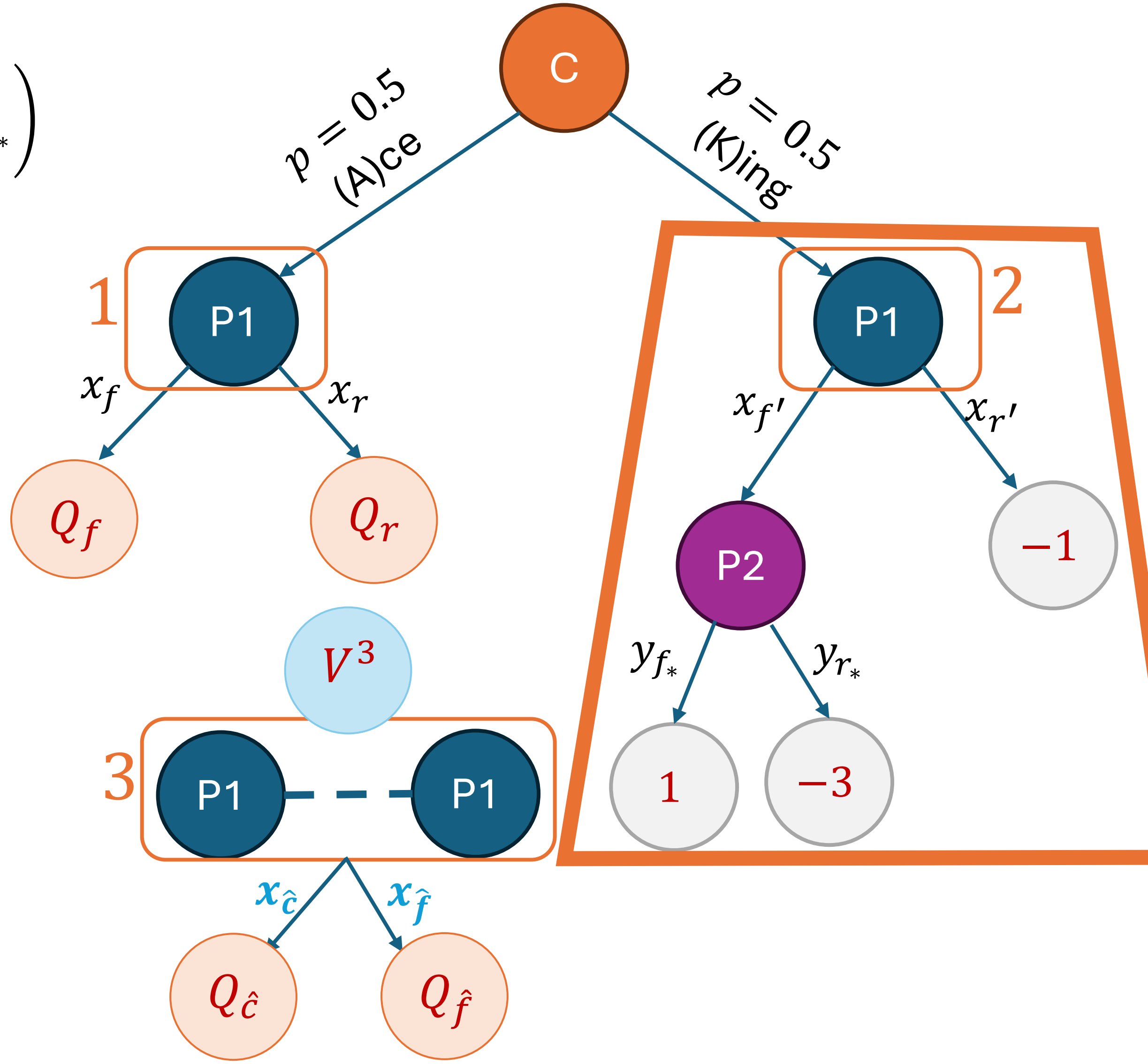


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= \left(u_{\hat{c}}, u_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, Q^3\right)$$

$$V^3 = \mathrm{softmax}(\eta_3 Q^3)$$

- Go to **Infoset 1**

$$U^1 \mathrel{+}= \left(u_f, u_r\right) = \left(-1\frac{1}{2}, 0\right)$$

$$Q^1 = U^1 + \left(0, V^3\right) = \left(-1\frac{1}{2}, V^3\right)$$

$$x^1 = \left(x_f, x_r\right) \propto \exp\left(\eta_1\, Q^1\right)$$

- Go to **Infoset 2**

$$U^2 \mathrel{+}= \left(u_{f'}, u_{r'}\right) = \left(1\frac{1}{2}y_{f_*} - 3\frac{1}{2}y_{r_*}, -1\frac{1}{2}\right)$$

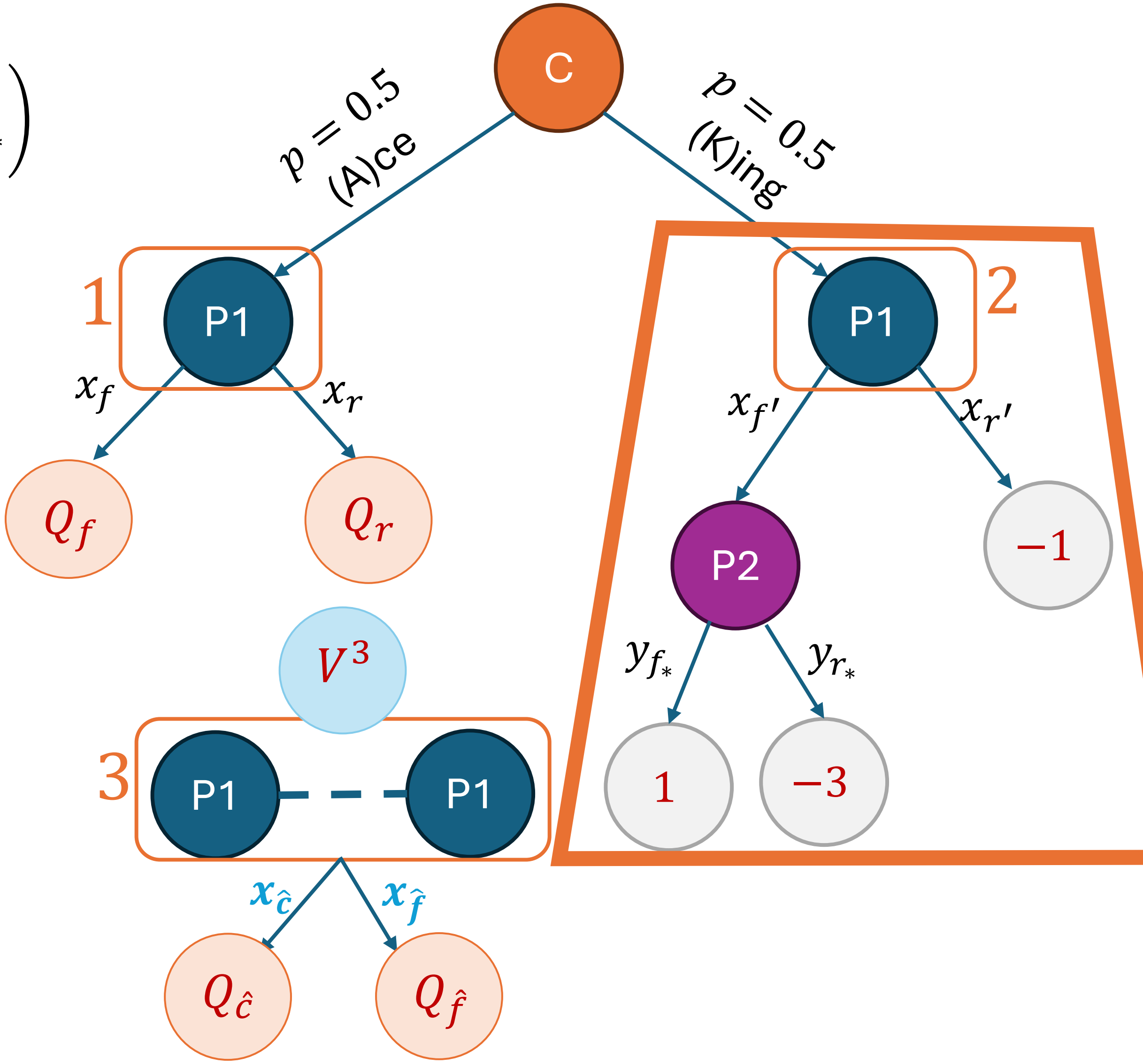


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$U^3 \mathrel{+}= (u_{\hat{c}}, u_{\hat{f}}) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$Q^3 = U^3, \qquad x^3 = (x_{\hat{c}}, x_{\hat{f}}) \propto \exp(\eta_3\, Q^3)$$

$$V^3 = \text{softmax}(\eta_3 Q^3)$$

- Go to **Infoset 1**

$$U^1 \mathrel{+}= (u_f, u_r) = \left(-1\frac{1}{2}, 0\right)$$

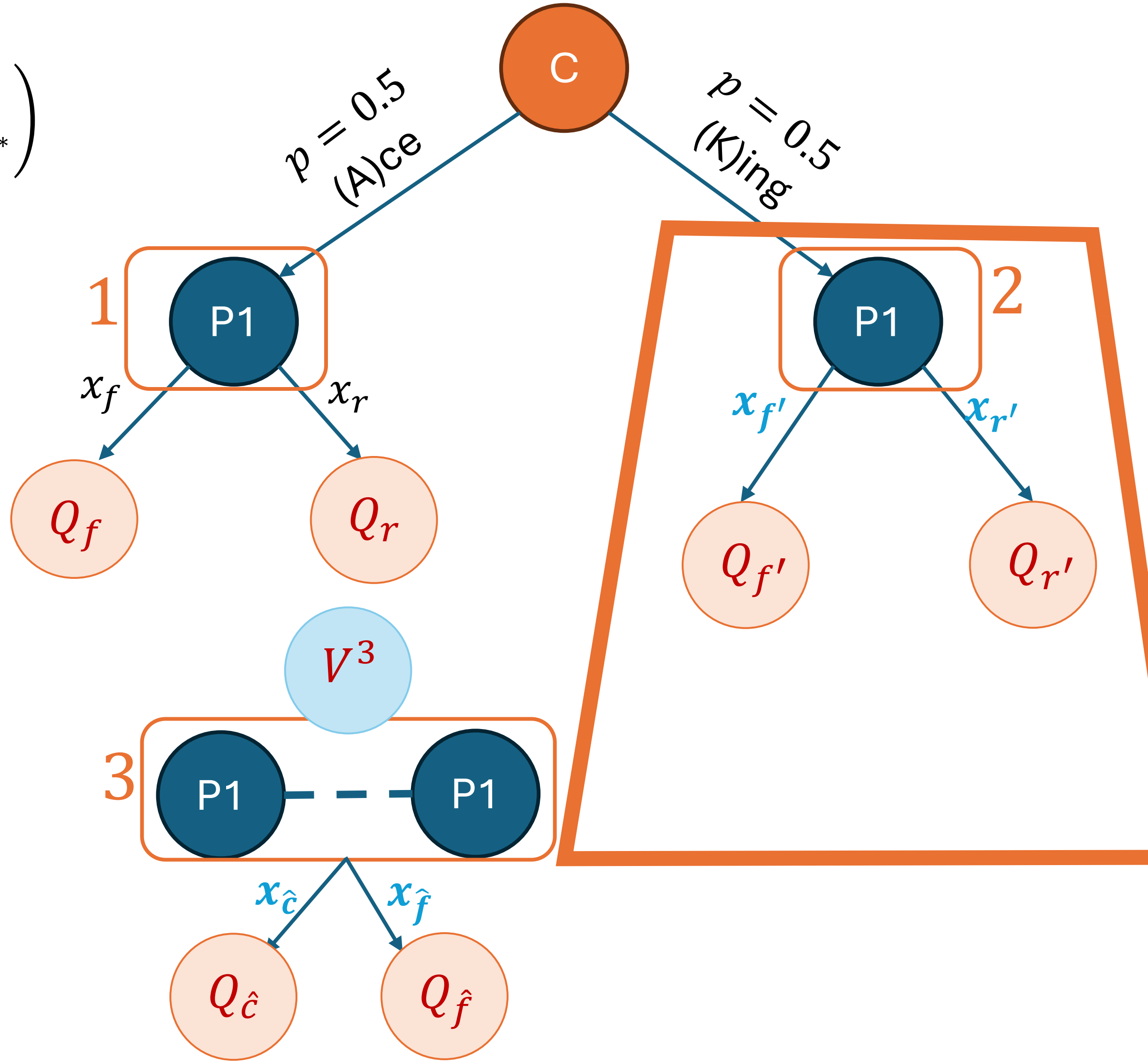
$$Q^1 = U^1 + (0, V^3) = \left(-1\frac{1}{2}, V^3\right)$$

$$x^1 = (x_f, x_r) \propto \exp(\eta_1\, Q^1)$$

- Go to **Infoset 2**

$$U^2 \mathrel{+}= (u_{f'}, u_{r'}) = \left(1\frac{1}{2}y_{f_*} - 3\frac{1}{2}y_{r_*}, -1\frac{1}{2}\right)$$

$$Q^2 = U^2, \qquad x^2 = (x_{f'}, x_{r'}) \propto \exp(\eta_2 Q^2)$$



Sum: Nash via FTRL with Dilated Entropy

Each player chooses $\tilde{x}_t, \tilde{y}_t$ based on FTRL with dilated entropy

- For x-player $u_t = A\tilde{y}_t$ and $U_t = U_{t-1} + u_t$ and initialize $Q = U_t$
- Traverse the tree bottom-up; for each infoset $j \in \mathcal{J}_1$
  $$x_{t+1}^j \propto \exp\left(\eta_j Q^j\right), \qquad V^j = \mathrm{softmax}_{\eta_j}\left(Q^j\right), \qquad Q_{p_j} \leftarrow Q_{p_j} + V^j$$
- Define sequence-form strategies top-down: $\tilde{x}_{t+1}^j = \tilde{x}_{p_j} \cdot x_{t+1}^j$

Similarly, for y player

Return average of sequence-form strategies as equilibrium

Fast Rates

Theorem. If we use Optimistic FTRL instead of FTRL then we get faster convergence to a Nash equilibrium at rate $1/T$ instead of $1/\sqrt{T}$. Plus, we get last-iterate convergence instead of only average iterate convergence.

Monte-Carlo Stochastic Approximation of Utilities

- Calculating utilities on all nodes of the tree can be very expensive
- In linear online learning it suffices that we use an unbiased estimate of the utility vector

All random variables observed before period τ

$$\tilde{x}_t = \operatorname*{argmax}_{x \in X} \sum_{\tau < t} \langle x, \hat{u}_\tau \rangle - \frac{1}{\eta} \mathcal{R}(x), \qquad E[\,\hat{u}_\tau \mid \boxed{F_\tau}\,] = u_\tau$$

- By standard martingale concentration inequality arguments, the error vanishes with the number of iterations (*we will see later*)
- In this setting, it suffices that we "sample a path for opponent" and that we "sample chance moves"

Illustration: First Step of Dynamics

- Sample chance moves based on fixed distribution and opponent moves based on y_t; Suppose, we sampled A and f_*

- Go to **Infoset 3**

$$\widehat{U}^3 \mathrel{+}= \left(\hat{u}_{\hat{c}}, \hat{u}_{\hat{f}}\right) = (-3, -2)$$

$$\widehat{Q}^3 = \widehat{U}^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3\, \widehat{Q}^3\right)$$

$$\widehat{V}^3 = \operatorname{softmax}(\eta_3 \widehat{Q}^3)$$

- Go to **Infoset 1**

$$\widehat{U}^1 \mathrel{+}= \left(\hat{u}_f, \hat{u}_r\right) = (-1, 0)$$

$$\widehat{Q}^1 = \widehat{U}^1 + \left(0, \widehat{V}^3\right) = \left(-1, \widehat{V}^3\right)$$

$$x^1 = \left(x_f, x_r\right) \propto \exp\left(\eta_1\, \widehat{Q}^1\right)$$

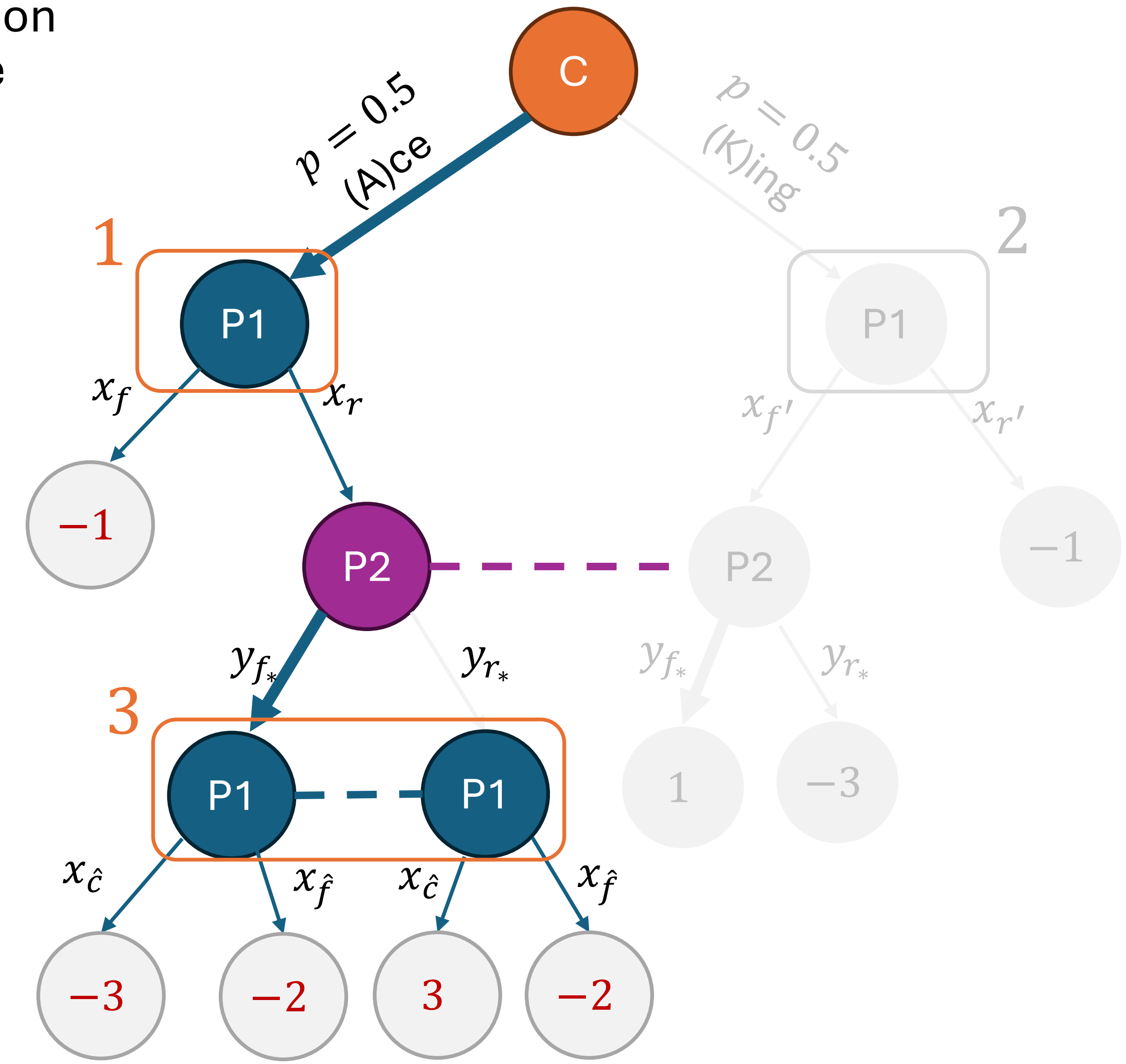


Illustration: First Step of Dynamics

Equivalently top down and evaluate recursively

- Sample chance move (e.g. sampled A)
- Go to **Infoset 1**

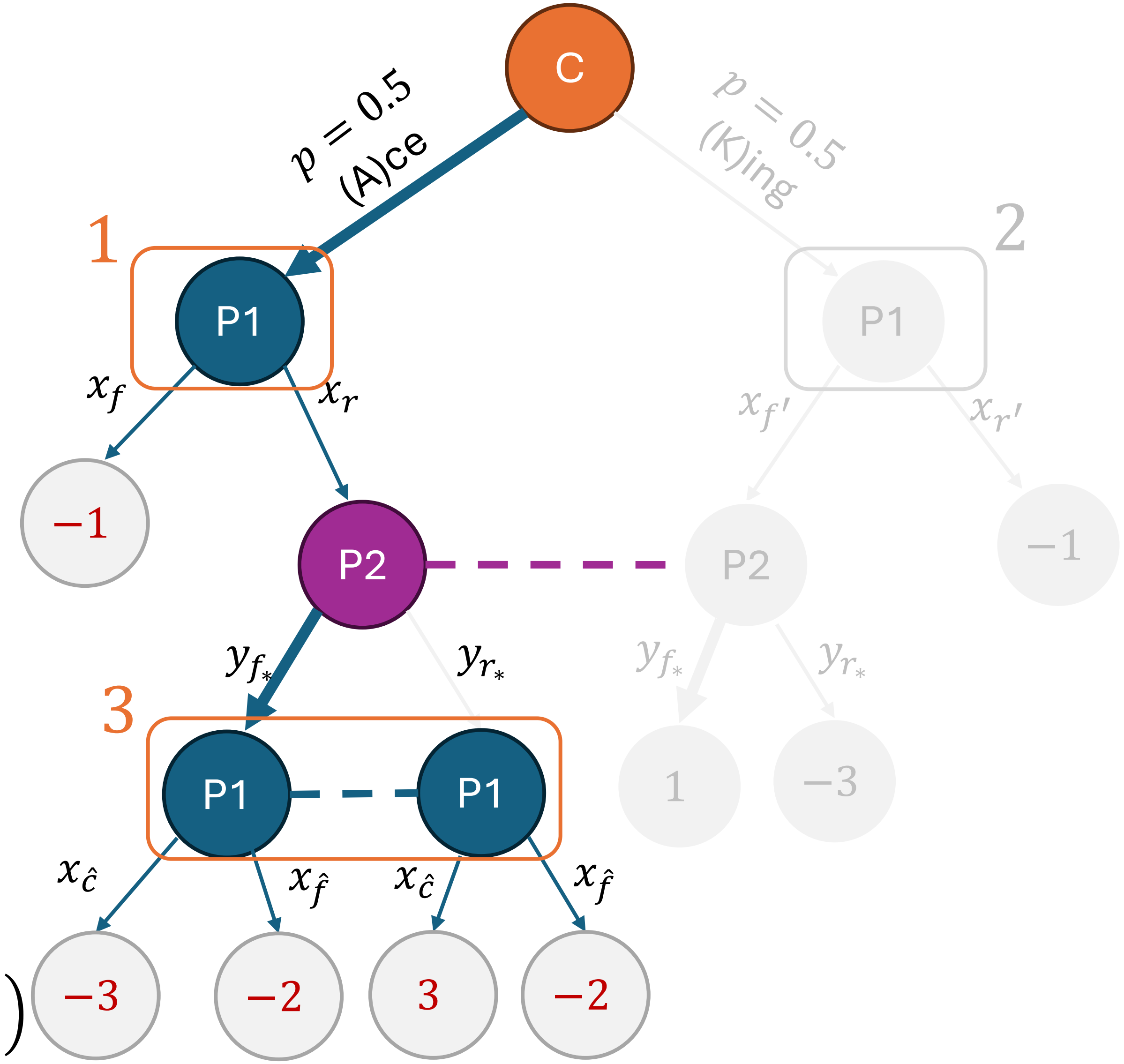
$$\hat{U}^1 \coloneqq (\hat{U}_f, \hat{U}_r) \mathrel{+}= (-1, 0)$$
$$\hat{Q}^1 \coloneqq (\hat{Q}_f, \hat{Q}_r) = (\hat{U}_f, \hat{U}_r)$$

- Recursively go down tree after action r
- Sample P2 move (e.g. sampled f_*)
- Go down to **Infoset 3**

$$\hat{U}^3 = \left(\hat{U}_{\hat{c}}, \hat{U}_{\hat{f}}\right) \mathrel{+}= (-3, -2)$$
$$\hat{Q}^3 = \hat{U}^3, \qquad x^3 = \left(x_{\hat{c}}, x_{\hat{f}}\right) \propto \exp\left(\eta_3 \, \hat{Q}^3\right)$$
$$\hat{V}^3 = \mathrm{softmax}(\eta_3 \hat{Q}^3)$$

- Go back up to **Infoset 1**

$$\hat{Q}_r \mathrel{+}= \hat{V}^3, \qquad x^1 = (x_f, x_r) \propto \exp\left(\eta_1 \left(\hat{Q}_f, \hat{Q}_r\right)\right)$$



Local Dynamics

- These dynamics seem to be doing “local updates” at each node
- They came out of a specific algorithm FTRL with Dilated Entropy
- Is this a general paradigm?
- Can we decompose the no-regret learning problem into local no-regret learners at each node?
- What feedback should each node receive from the learners in nodes below?
- What loss should each learner be optimizing?

Counterfactual Regret Minimization (CRM)

Re-interpretating Utilities

Interpretation of u_a. If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then don't make any other moves*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

What if we now want to express: If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then continue playing based on some behavioral policy x*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

Re-interpretating Utilities

Interpretation of u_a. If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then don't make any other moves*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

What if we now want to express: If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then continue playing based on some behavioral policy x*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

- Let C_a be all infosets of the player that are reachable **as next infosets** after playing a

$$\tilde{u}_a(x) = \boxed{u_a} + \sum_{k \in C_a} \boxed{V^k(x)}$$

"Instantaneous E[utility]", if this is the last action I play

Continuation E[utility] from paths that pass through infoset k, if I continue playing based on behavioral strategy x

Re-interpretating Utilities

Interpretation of $\boldsymbol{u_a}$. If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then don't make any other moves*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

What if we now want to express: If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then continue playing based on some behavioral policy x*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

- Let C_a be all infosets of the player that are reachable **as next infosets** after playing a

$$\tilde{u}_a(x) = \underbrace{u_a}_{\text{"Instantaneous E[utility]", if this is the last action I play}} + \sum_{k \in C_a} \underbrace{V^k(x)}_{\text{Continuation E[utility] from paths that pass through infoset } k \text{, if I continue playing based on behavioral strategy } x}$$

- Continuation utility $V^j(x)$ from paths that pass through infoset j recursively defined:

$$V^j(x) = \sum_{a \in A^j} x_a \, \tilde{u}_a(x) = \underbrace{\sum_{a \in A^j} x_a u_a}_{\text{"Instantaneous utility", if this is the last move I make}} + \underbrace{\sum_{a \in A^j} x_a \left(\sum_{k \in C_a} V^k(x) \right)}_{\text{"Continuation utility", if I continue playing based on } x}$$

Re-interpretating Utilities

- Continuation utility $V^j(x)$ from paths that pass through j, assuming I play to arrive deterministically at the parent action p_j (i.e., $\tilde{x}_{p_j} = 1$)

$$V^j(x) = \sum_{a \in A^j} x_a \, \tilde{u}_a(x) = \sum_{a \in A^j} x_a \left(u_a + \sum_{k \in C_a} V^k(x) \right)$$

- Obviously $V^{\mathrm{root}}(x)$ is total expected utility from behavior strategy x
- From equivalence of behavioral and sequence-form strategies

$$V^{\mathrm{root}}(x) = \langle \tilde{x}, u \rangle$$

- The same also holds for regrets

$$R^{\mathrm{root}}(x) = \max_{x'} V^{\mathrm{root}}(x') - V^{\mathrm{root}}(x) = \max_{\tilde{x}' \in X} \langle \tilde{x}', u \rangle - \langle \tilde{x}, u \rangle = R(\tilde{x})$$

Local Regrets

- We can also define infoset regrets based on local utilities $\tilde{u}_a$

$$R^j(x) = \max_{x'} V^j(x') - V^j(x) = \max_{x'} \sum_a x'_a \tilde{u}_a(x') - x_a \tilde{u}_a(x)$$

- Right-hand-side can be decomposed as:

$$\max_{x'} \boxed{\sum_a x'_a \tilde{u}_a(x) - x_a \tilde{u}_a(x)} + \boxed{\sum_a x'_a \big(\tilde{u}_a(x') - \tilde{u}_a(x)\big)}$$

Fix continuation strategy to current strategy and only change the behavioral strategy at the current infoset

Weighted average of changes in continuation strategy

Local Regrets

- We can also define infoset regrets based on local utilities $\tilde{u}_a$

$$R^j(x) = \max_{x'} V^j(x') - V^j(x) = \max_{x'} \sum_a x'_a \tilde{u}_a(x') - x_a \tilde{u}_a(x)$$

- Right-hand-side can be decomposed as:

$$\max_{x'} \sum_a x'_a \tilde{u}_a(x) - x_a \tilde{u}_a(x) + \sum_a x'_a \left(\tilde{u}_a(x') - \tilde{u}_a(x)\right)$$

- Maximum is upper bounded by the decoupled optima

$$\boxed{\max_{x'} \sum_a x'_a \tilde{u}_a(x) - x_a \tilde{u}_a(x)} + \sum_a \max_{x'} \left(\tilde{u}_a(x') - \tilde{u}_a(x)\right)$$

Local Regret:* LR^j*(x)

Regret if you only change current info set behavioral strategy and keep continuation strategy

Recursive Bound of Local Regrets

- Infoset regrets are bounded by local regret plus continuation terms

$$R^j(x) \leq \mathrm{LR}^j(x) + \sum_a \max_{x'}\left(\tilde{u}_a(x') - \tilde{u}_a(x)\right)$$

- The continuation terms are recursive infoset regrets!

$$\tilde{u}_a(x') - \tilde{u}_a(x) = \cancel{u_a} + \sum_{k \in C_a} V^k(x') - \cancel{u_a} - \sum_{k \in C_a} V^k(x)$$

- Deriving the recursive upper bound

$$R^j(x) \leq \mathrm{LR}^j(x) + \sum_a \sum_{k \in C_a} \max_{x'} V^k(x') - V^k(x)$$

$$\leq \mathrm{LR}^j(x) + \sum_a \sum_{k \in C_a} R^k(x)$$

Recursive Bound of Local Regrets

- Deriving the recursive upper bound

$$R^j(x) \le \mathrm{LR}^j(x) + \sum_a \sum_{k \in C_a} R^k(x)$$

Recursive Bound of Local Regrets

- Deriving the recursive upper bound

$$R^j(x) \leq \mathrm{LR}^j(x) + \sum_a \sum_{k \in C_a} R^k(x)$$

Theorem. By induction:

$$R^j(x) \leq LR^j(x) + \sum_{k \text{ eventually reachable from } j} LR^k(x)$$

Local Regrets Upper Bound Total Regret

- Deriving the recursive upper bound

$$R^j(x) \leq \mathrm{LR}^j(x) + \sum_a \sum_{k \in C_a} R^k(x)$$

Theorem. By induction:

$$R^j(x) \leq LR^j(x) + \sum_{k \text{ eventually reachable from } j} LR^k(x)$$

Main Corollary. Regret is upper bounded by sum of local regrets

$$R(\tilde{x}) = R^{\mathrm{root}}(x) \leq \sum_{k \in \mathcal{J}_1} LR^k(x)$$

Regret over Time

Same inequalities can be followed for the average regret over time

$$R = \max_{\tilde{x}' \in X} \frac{1}{T} \sum_t \langle \tilde{x}', u_t \rangle - \langle \tilde{x}_t, u_t \rangle$$

$$LR^j = \max_{x^j} \frac{1}{T} \sum_t \langle x^j, \tilde{u}_t(x_t) \rangle - \left\langle x_t^j, \tilde{u}_t(x_t) \right\rangle$$

Main CFR Theorem. Regret is upper bounded by local regrets

$$R \leq \sum_{j \in \mathcal{L}_1} LR^j$$

Achieving vanishing Local Regrets

$$\mathrm{LR}^j(x) = \max_{x^j} \frac{1}{T} \sum_t \langle x^j, \widetilde{u}_t(x_t) \rangle - \left\langle x_t^j, \widetilde{u}_t(x_t) \right\rangle$$

Counterfactual Regret Minimization

- Device local regret algorithms for local regret

$$\mathrm{LR}^j(x) = \max_{x^j} \frac{1}{T} \sum_t \langle x^j, \tilde{u}_t(x_t) \rangle - \left\langle x_t^j, \tilde{u}_t(x_t) \right\rangle$$

- Standard n-action no-regret problem: reward vector at period t is $\tilde{u}^j(x_t)$ and reward for choice x^j is $\langle x^j, \tilde{u}^j(x_t) \rangle$

- At period t run bottom-up recursion to calculate $\tilde{u}^j(x_t)$ for $j \in \mathcal{J}_1$
- Update probabilities x_{t+1}^j using reward vectors $\tilde{u}^j(x_t)$ for $j \in \mathcal{J}_1$

Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

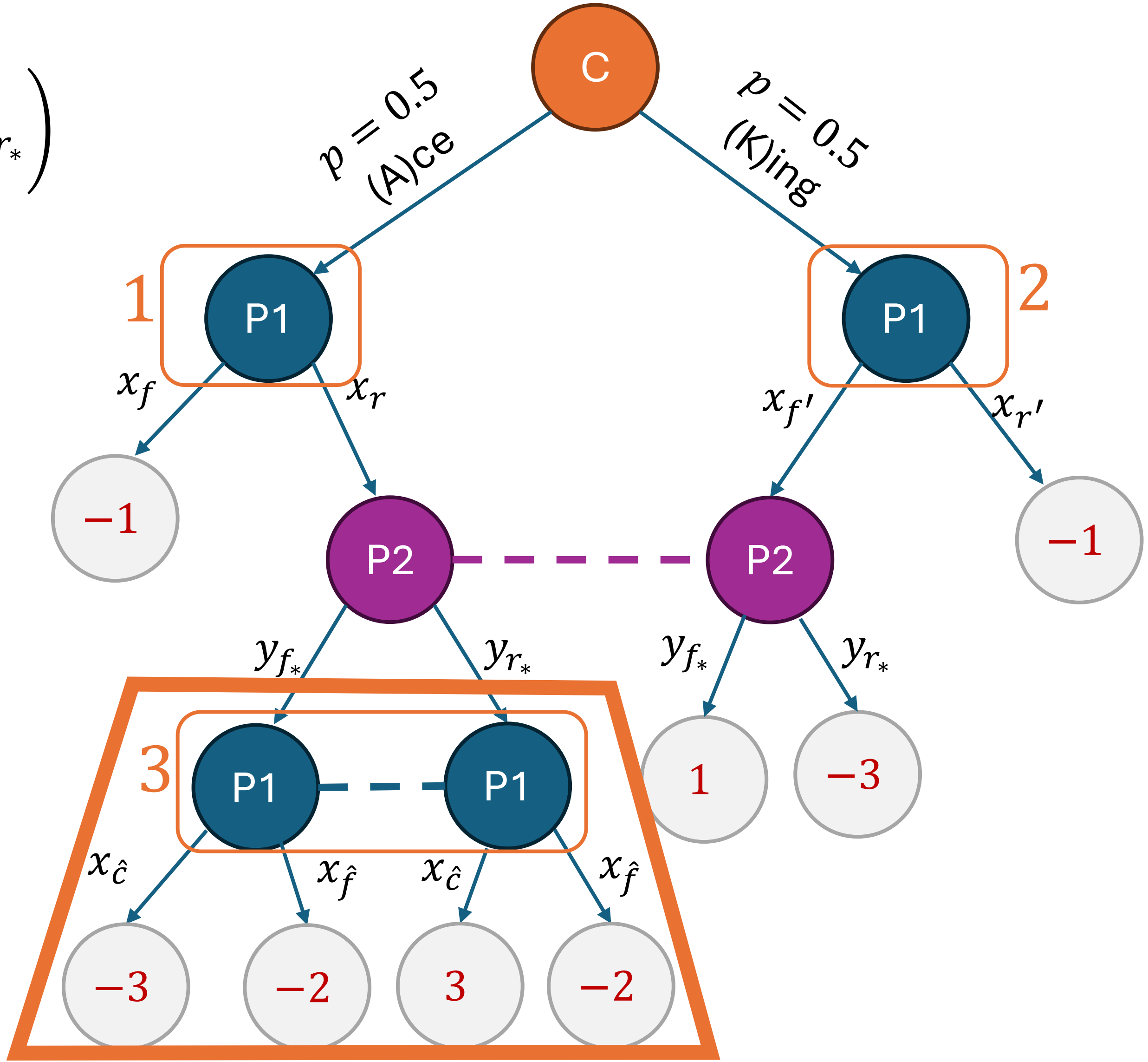


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

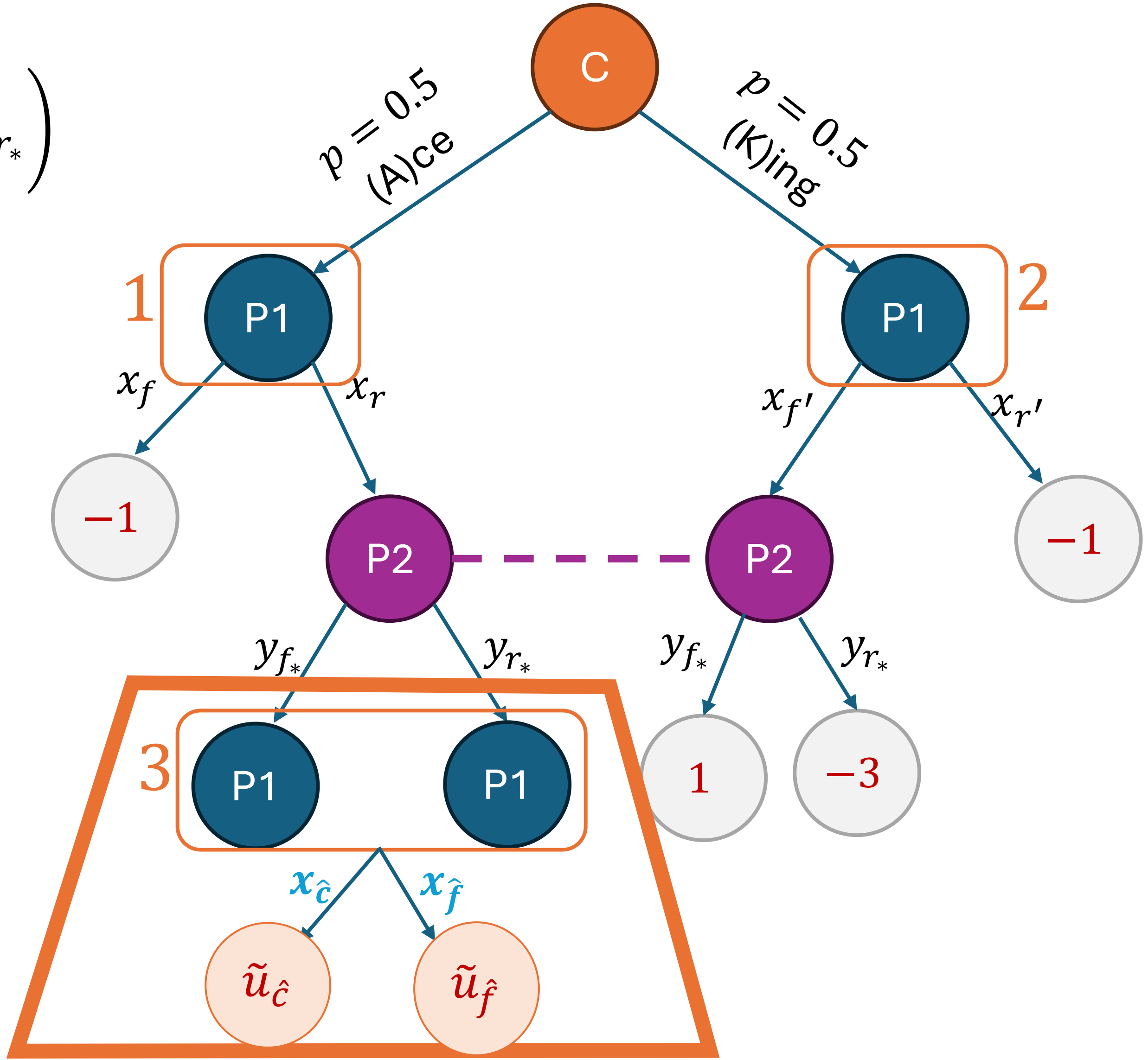


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$V^3 \leftarrow x_{\hat{c}}\tilde{u}_{\hat{c}} + x_{\hat{f}}\tilde{u}_{\hat{f}}$$

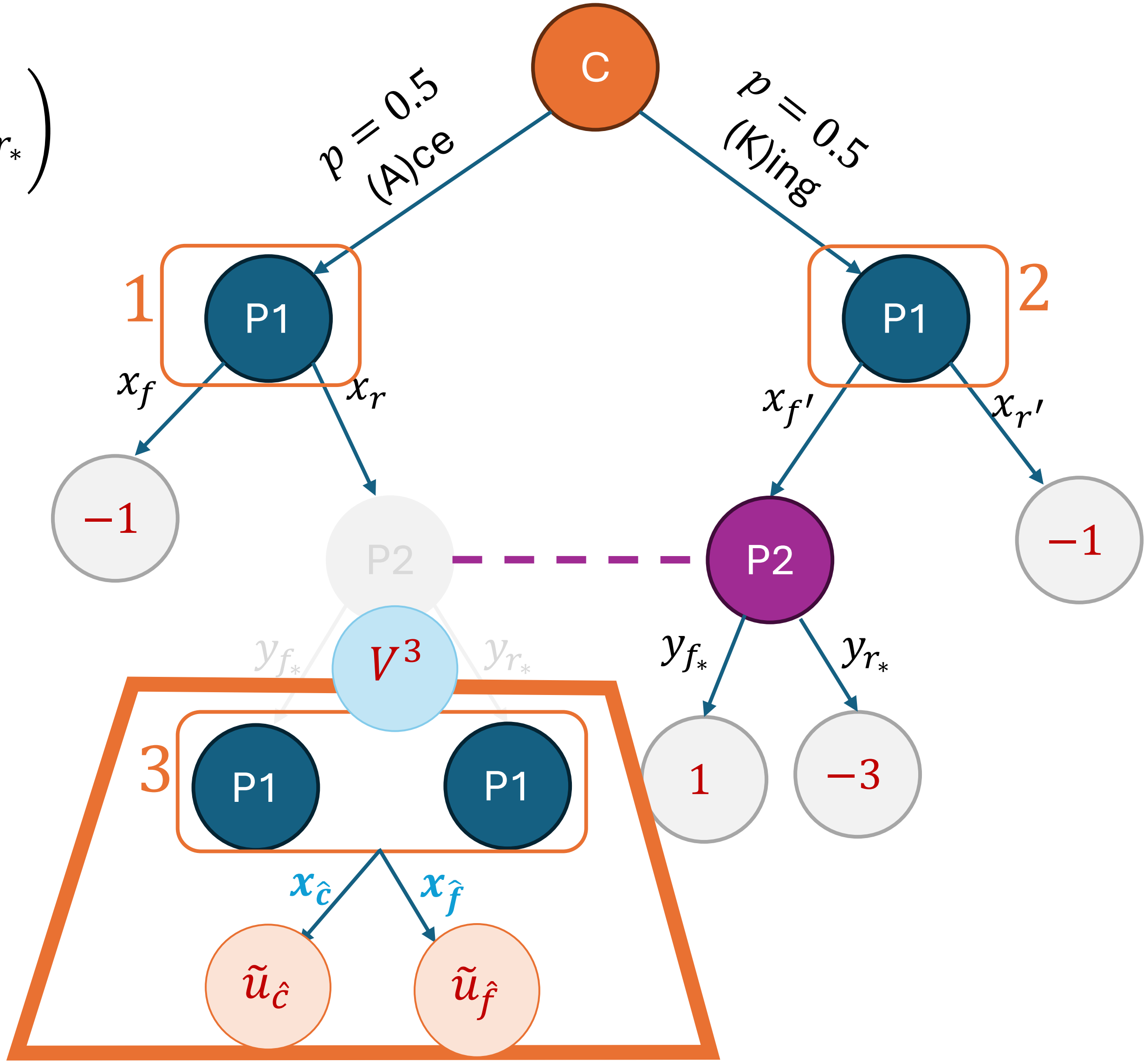


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$V^3 \leftarrow x_{\hat{c}}\tilde{u}_{\hat{c}} + x_{\hat{f}}\tilde{u}_{\hat{f}}$$

- Go to **Infoset 1**

$$\left(\tilde{u}_f, \tilde{u}_r\right) = \left(-1\frac{1}{2}, V^3\right)$$

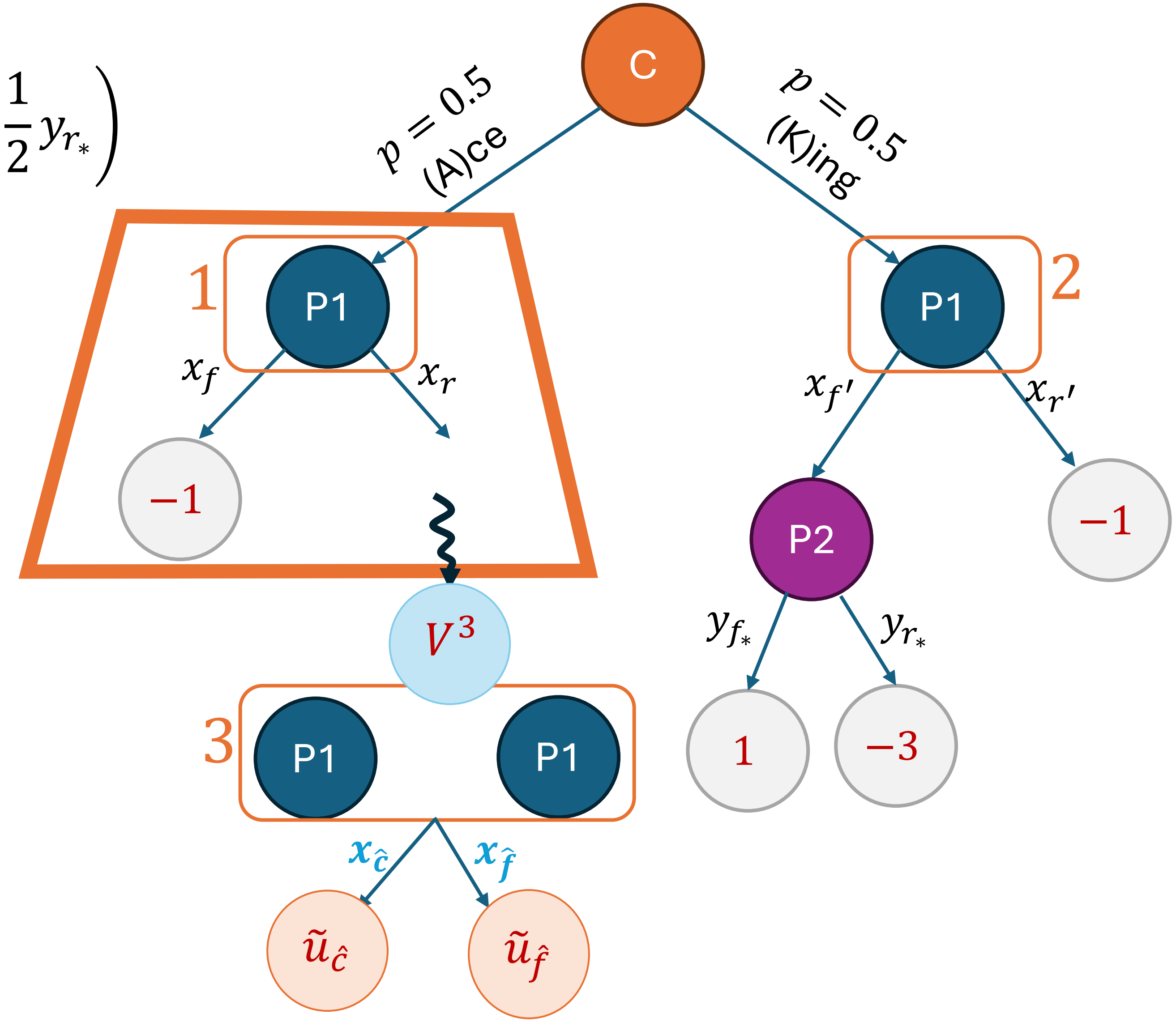


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$V^3 \leftarrow x_{\hat{c}}\tilde{u}_{\hat{c}} + x_{\hat{f}}\tilde{u}_{\hat{f}}$$

- Go to **Infoset 1**

$$\left(\tilde{u}_f, \tilde{u}_r\right) = \left(-1\frac{1}{2}, V^3\right)$$

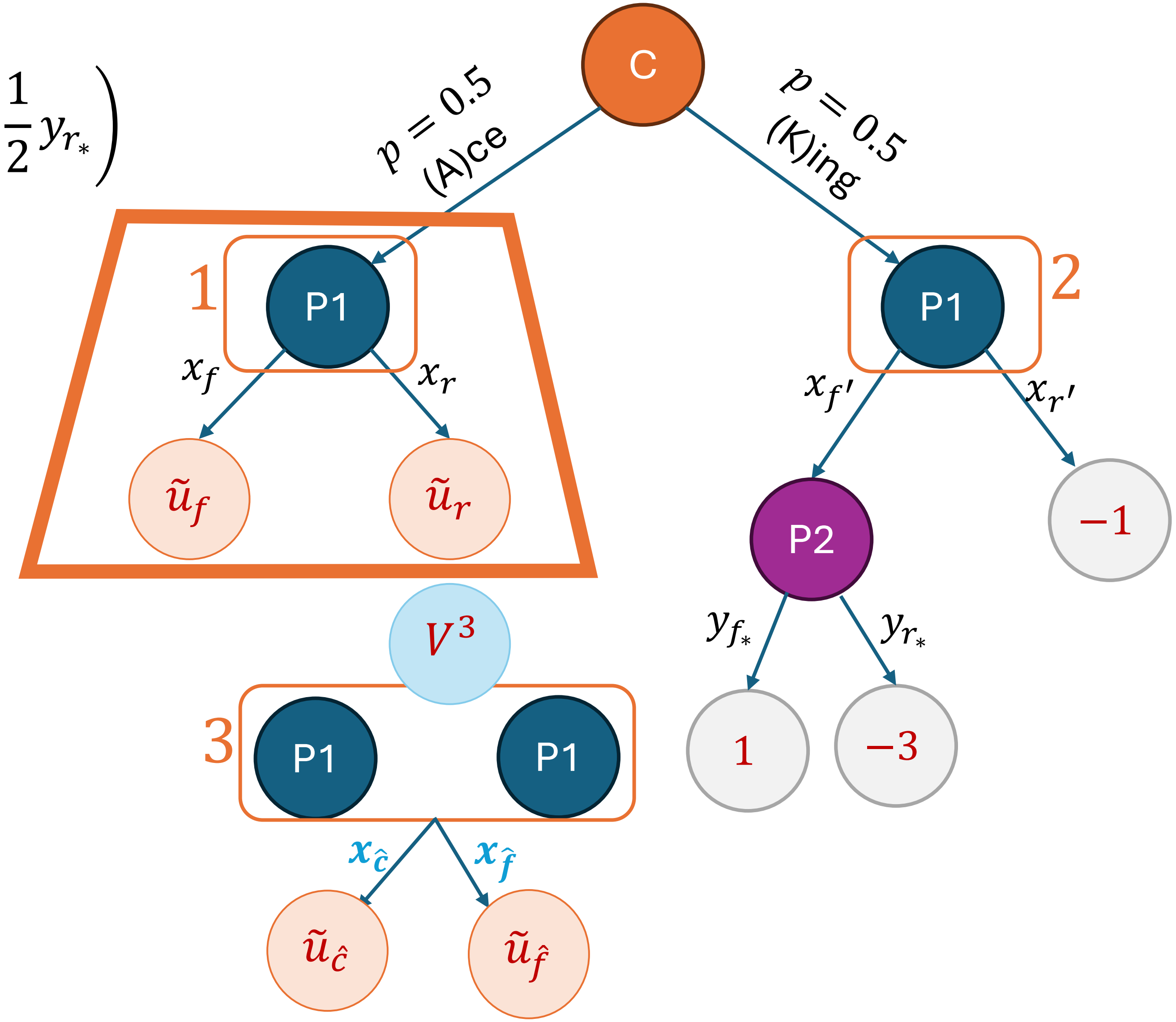


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$V^3 \leftarrow x_{\hat{c}}\tilde{u}_{\hat{c}} + x_{\hat{f}}\tilde{u}_{\hat{f}}$$

- Go to **Infoset 1**

$$\left(\tilde{u}_f, \tilde{u}_r\right) = \left(-1\frac{1}{2}, V^3\right)$$

- Go to **Infoset 2**

$$\left(\tilde{u}_{f'}, \tilde{u}_{r'}\right) = \left(1\frac{1}{2}y_{f_*} - 3\frac{1}{2}y_{r_*}, -1\frac{1}{2}\right)$$

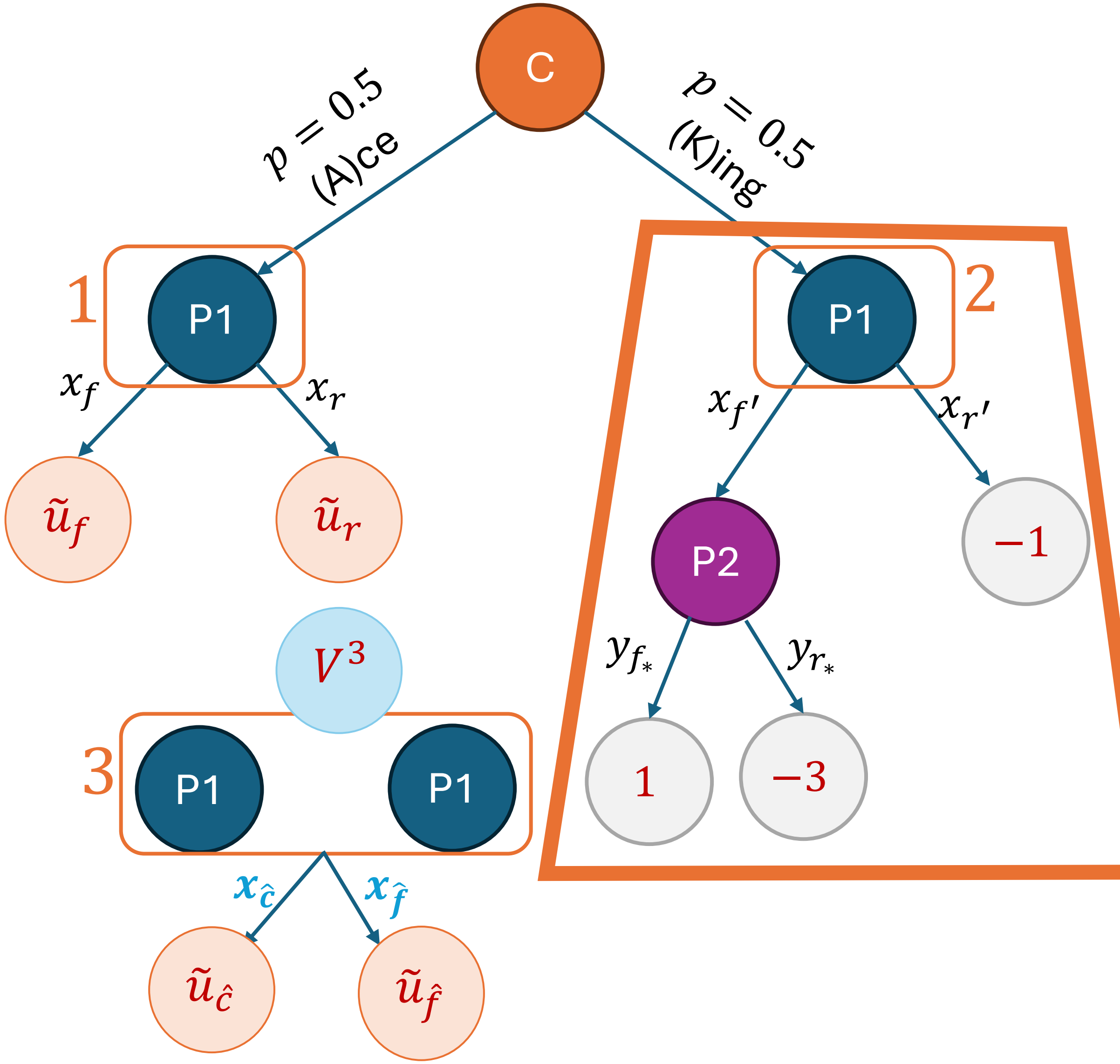


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$V^3 \leftarrow x_{\hat{c}}\tilde{u}_{\hat{c}} + x_{\hat{f}}\tilde{u}_{\hat{f}}$$

- Go to **Infoset 1**

$$\left(\tilde{u}_f, \tilde{u}_r\right) = \left(-1\frac{1}{2}, V^3\right)$$

- Go to **Infoset 2**

$$\left(\tilde{u}_{f'}, \tilde{u}_{r'}\right) = \left(1\frac{1}{2}y_{f_*} - 3\frac{1}{2}y_{r_*}, -1\frac{1}{2}\right)$$

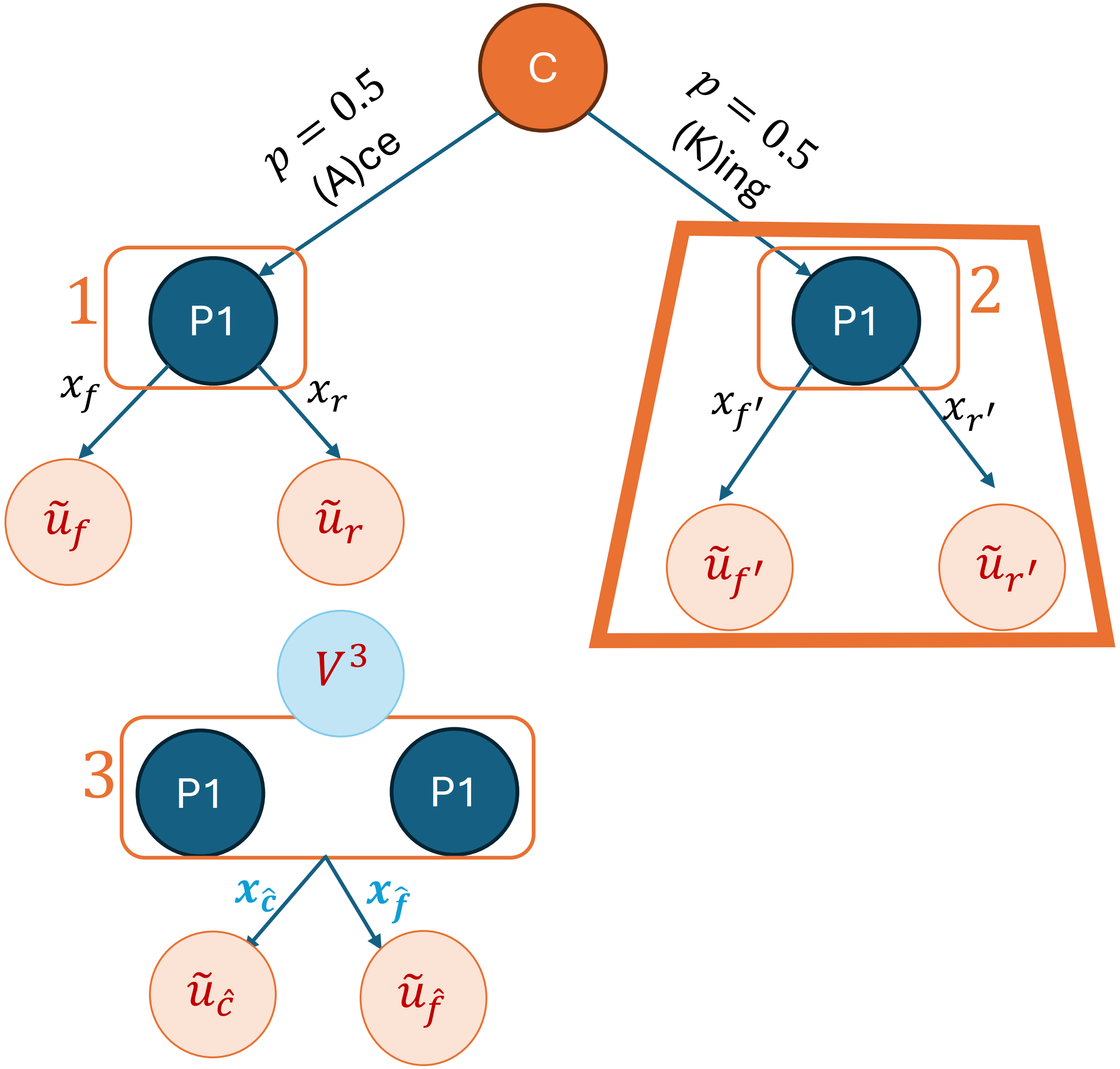


Illustration: First Step of Dynamics

- Go to **Infoset 3**

$$\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right) = \left(-3\frac{1}{2}y_{f_*} + 3\frac{1}{2}y_{r_*}, -2\frac{1}{2}y_{f_*} - 2\frac{1}{2}y_{r_*}\right)$$

$$V^3 \leftarrow x_{\hat{c}}\tilde{u}_{\hat{c}} + x_{\hat{f}}\tilde{u}_{\hat{f}}$$

- Go to **Infoset 1**

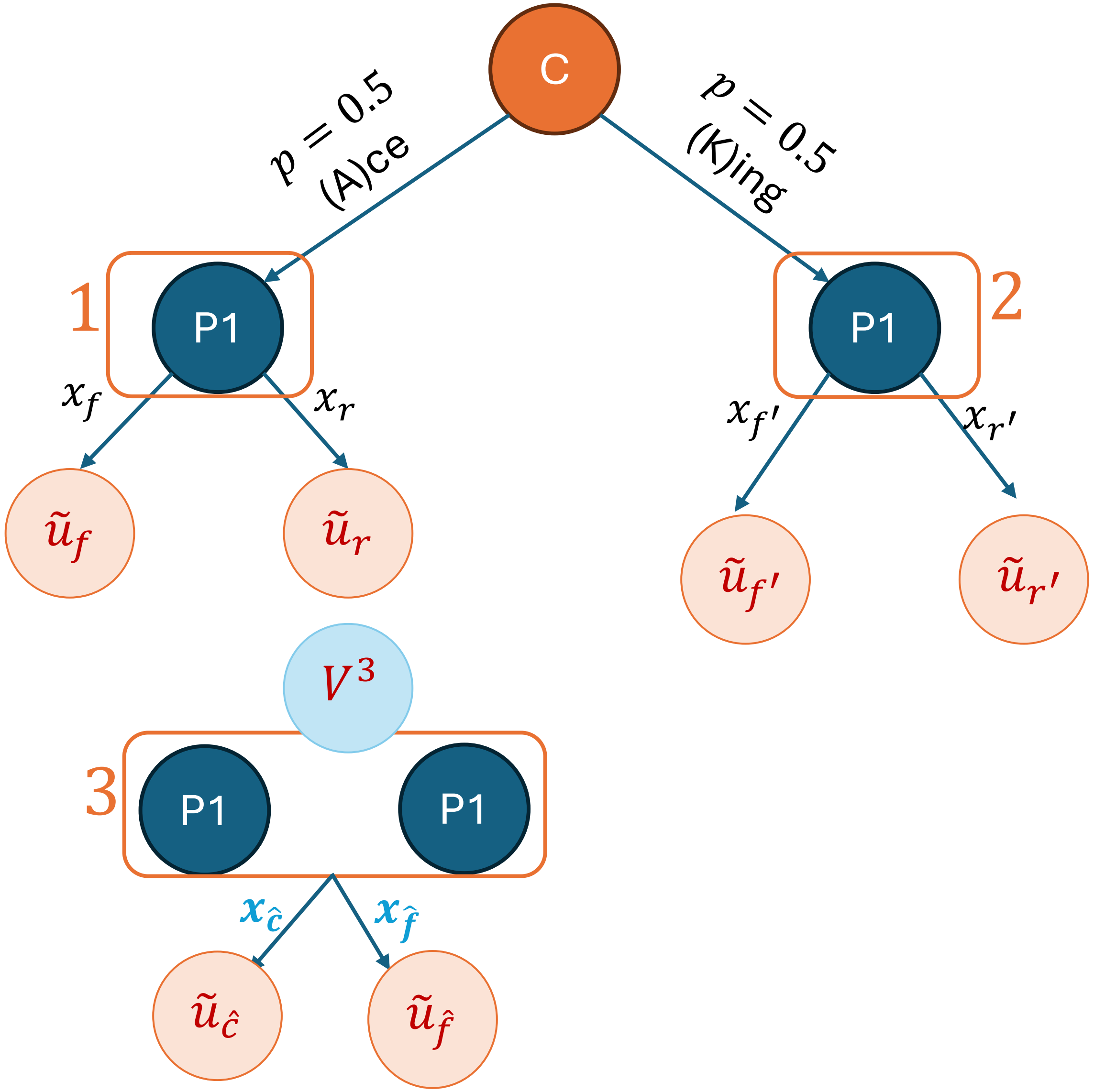
$$\left(\tilde{u}_f, \tilde{u}_r\right) = \left(-1\frac{1}{2}, V^3\right)$$

- Go to **Infoset 2**

$$\left(\tilde{u}_{f'}, \tilde{u}_{r'}\right) = \left(1\frac{1}{2}y_{f_*} - 3\frac{1}{2}y_{r_*}, -1\frac{1}{2}\right)$$

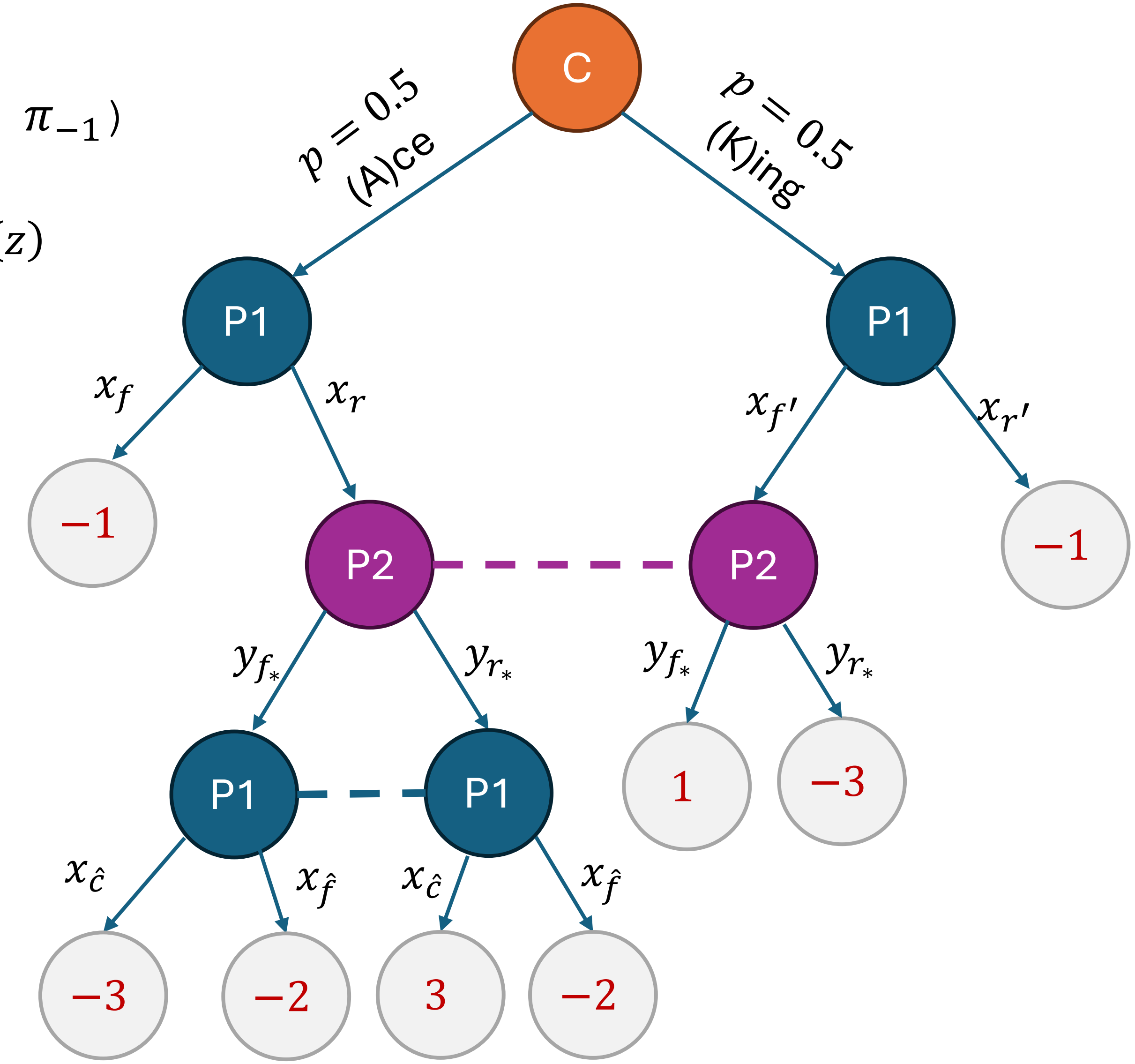
- **Update probabilities**

$$\left(x_f, x_r\right) \leftarrow \mathrm{Update}\left(\tilde{u}_f, \tilde{u}_r\right)$$
$$\left(x_{f'}, x_{r'}\right) \leftarrow \mathrm{Update}\left(\tilde{u}_{f'}, \tilde{u}_{r'}\right)$$
$$\left(x_{\hat{c}}, x_{\hat{f}}\right) \leftarrow \mathrm{Update}\left(\tilde{u}_{\hat{c}}, \tilde{u}_{\hat{f}}\right)$$



Recursive Algorithm

```
Value(ActionHistory h, AccOtherProb π₋₁)
    Let I be infoset corresponding to h
    If I is terminal node z return π₋₁·u(z)
    If Player(I) = chance
        Return Σ_{a∈A_I} Value(ha, π₋₁π_a^C)
    If Player(I) = 2
        Return Σ_{a∈A_I} Value(ha, π₋₁y_a)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += Value(ha, π₋₁)
        Return Σ_{a∈A_I} x_a · Value(ha, π₋₁)

Value(∅, 1)
```

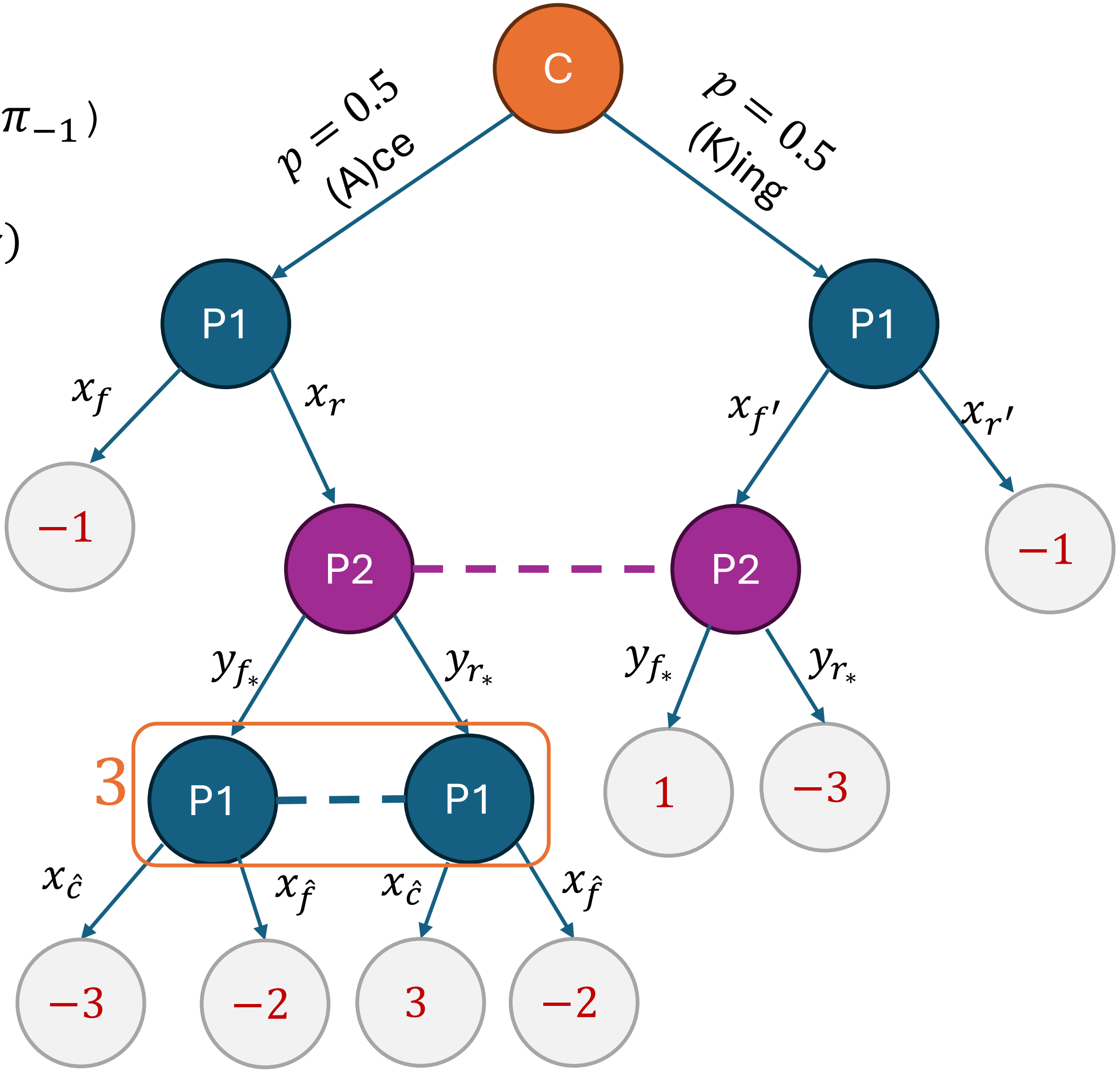


Recursive Algorithm

```
Value(ActionHistory h, AccOtherProb π_{-1})
    Let I be infoset corresponding to h
    If I is terminal node z return π_{-1}·u(z)
    If Player(I) = chance
        Return Σ_{a∈A_I} Value(ha, π_{-1}π^C_a)
    If Player(I) = 2
        Return Σ_{a∈A_I} Value(ha, π_{-1}y_a)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += Value(ha, π_{-1})
        Return Σ_{a∈A_I} x_a · Value(ha, π_{-1})
```

We arrive at the same infoset I multiple times, once for each node in the set; $\tilde{u}_a$ accumulates continuation utility from taking action a from all these possible "arrival paths".

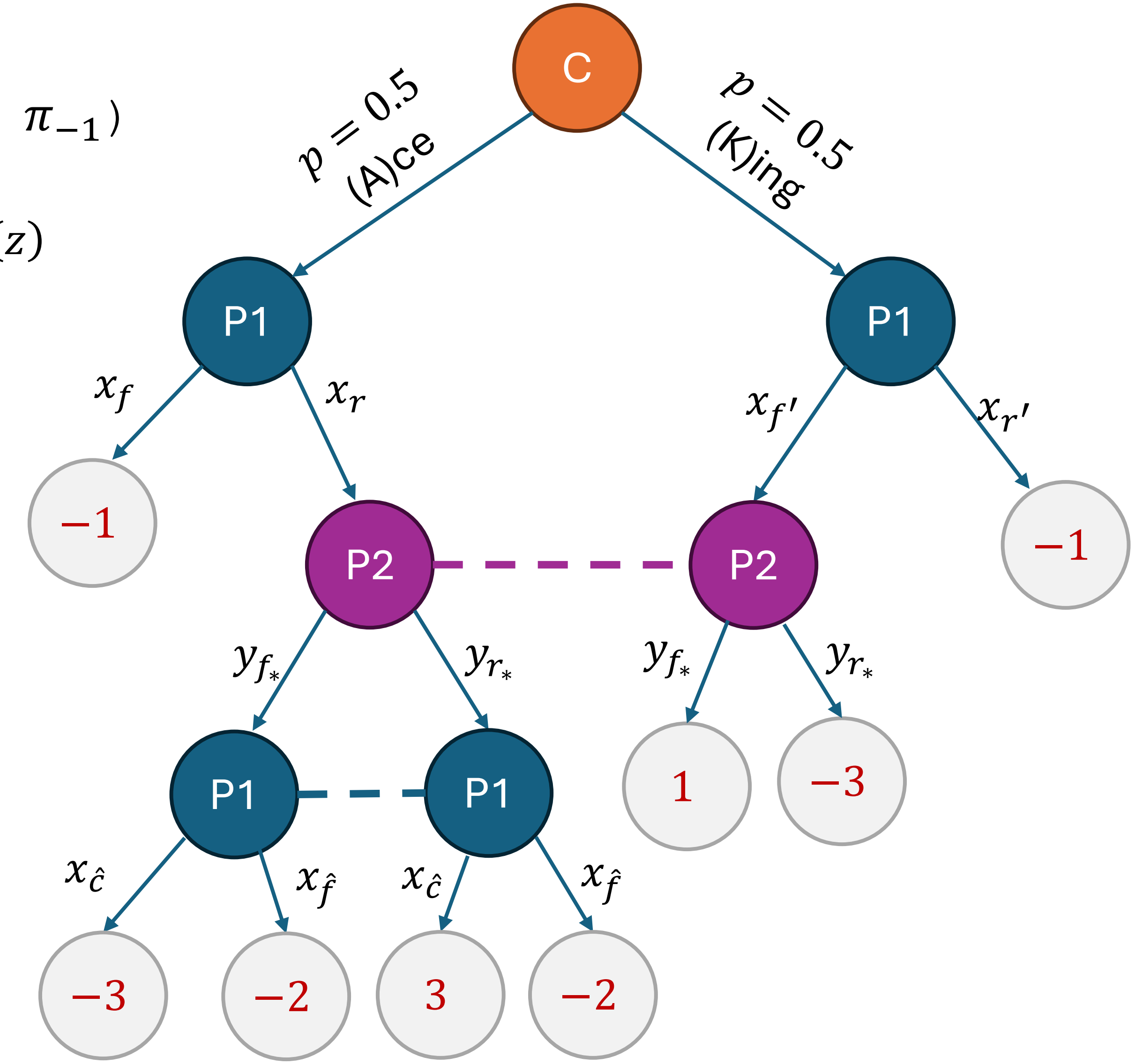
Example. In infoset 3 we arrive once on the left node and add $-3\frac{1}{2}y_{f_*}$ and once on the right node and add $3\frac{1}{2}y_{r_*}$ to $u_{\hat{c}}$



Recursive Algorithm

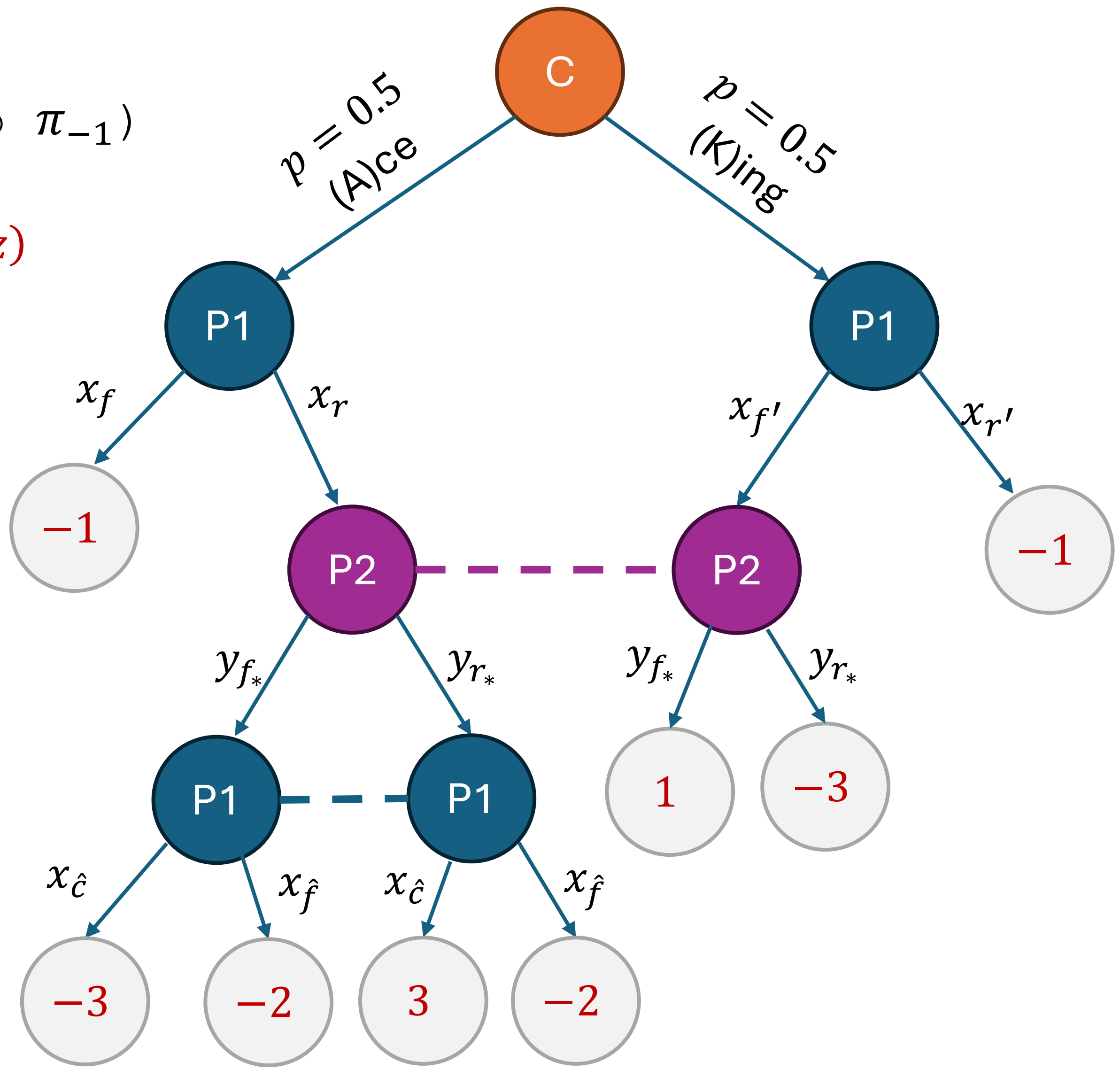
```
Value(ActionHistory h, AccOtherProb π_{-1})
    Let I be infoset corresponding to h
    If I is terminal node z return π_{-1} · u(z)
    If Player(I) = chance
        Return Σ_{a∈A_I} Value(ha, π_{-1} π_a^C)
    If Player(I) = 2
        Return Σ_{a∈A_I} Value(ha, π_{-1} y_a)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += Value(ha, π_{-1})
        Return Σ_{a∈A_I} x_a · Value(ha, π_{-1})
```

```
Value(∅, 1)
```



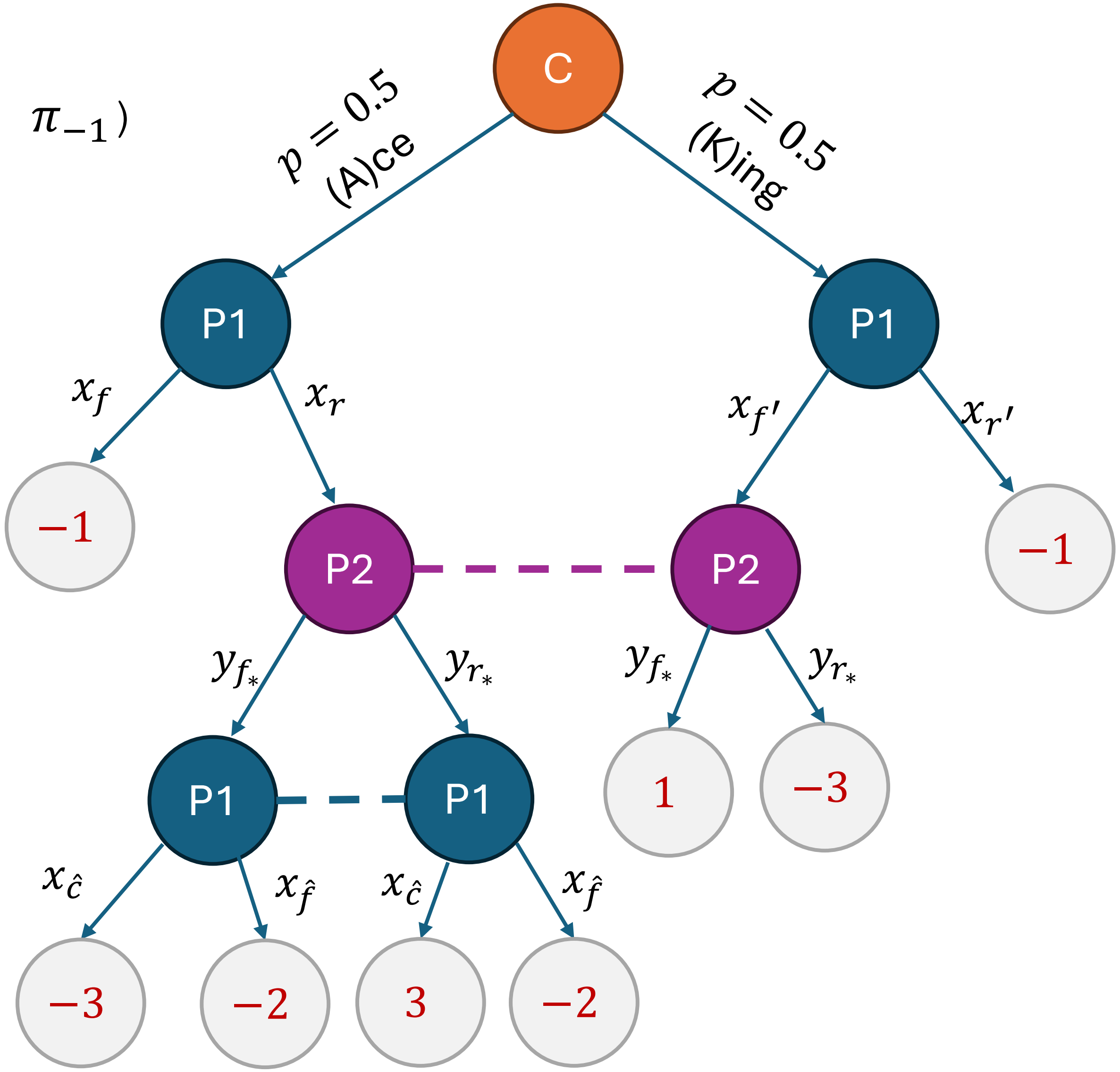
Equivalent Recursive Algorithm

```
CValue(ActionHistory h, AccOtherProb π₋₁)
    Let I be infoset corresponding to h
    If I is terminal node z return π₋₁ · u(z)
    If Player(I) = chance
        Return Σ_{a∈A_I} π_a^C · CValue(ha, π₋₁π_a^C)
    If Player(I) = 2
        Return Σ_{a∈A_I} y_a · CValue(ha, π₋₁y_a)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += π₋₁ · CValue(ha, π₋₁)
        Return Σ_{a∈A_I} x_a · CValue(ha, π₋₁)


CValue(∅, 1)
```



The Typical CRM Algorithm Implementation

```
CValue(ActionHistory h, AccOtherProb π_{-1})
    Let I be infoset corresponding to h
    If I is terminal node z return u(z)
    If Player(I) = chance
        Return Σ_{a∈A_I} π_a^C · CValue(ha, π_{-1} π_a^C)
    If Player(I) = 2
        Return Σ_{a∈A_I} y_a · CValue(ha, π_{-1} y_a)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += π_{-1} · CValue(ha, π_{-1})
        Return Σ_{a∈A_I} x_a · CValue(ha, π_{-1})


CValue(∅, 1)
```



Recovering Equilibrium from CRM Dynamics

We have run CRM dynamics generating behavioral strategies x_t, y_t for T periods.

How do we calculate the behavioral strategies x^*, y^* that are an approximate Nash equilibrium?

Recovering Nash Equilibrium

- We need to translate the behavioral strategies into sequence-form

$$\forall a \in A_j:\ \tilde{x}_{t,a} = \boxed{\tilde{x}_{t,p_j}} \cdot x_t$$

Product of probabilities of actions of player P1 on path to infoset of action i

- Then average the sequence-form strategies

$$\bar{\tilde{x}} = \frac{1}{T}\sum_{t=1}^{T} \tilde{x}_t$$

- Then translate back to equilibrium behavioral strategies x^*

$$\forall a \in A_j:\ x_a^* = \frac{\bar{\tilde{x}}_a}{\bar{\tilde{x}}_{p_j}}$$

Recovering Nash Equilibrium

- We need to translate the behavioral strategies into sequence-form

$$\forall a \in A_j:\ \tilde{x}_{t,a} = \boxed{\tilde{x}_{t,p_j}} \cdot x_t$$

Product of probabilities of actions of player P1 on path to infoset of action i

- Then average the sequence-form strategies

$$\bar{\tilde{x}} = \frac{1}{T}\sum_{t=1}^{T} \tilde{x}_t = \frac{1}{T}\sum_{t=1}^{T} \tilde{x}_{t,p_j} \cdot x_t$$

- Then translate back to equilibrium behavioral strategies x^*

$$\forall a \in A_j:\ x_a^* = \frac{\bar{\tilde{x}}_a}{\bar{\tilde{x}}_{p_j}} = \frac{\sum_{t=1}^{T} \tilde{x}_{t,p_j} \cdot x_{t,a}}{\sum_{t=1}^{T} \tilde{x}_{t,p_j}}$$

The Typical CRM Algorithm Implementation

```
CValue(ActionHistory h, AccOtherProb π₋₁, AccProb π₁)
    Let I be infoset corresponding to h
    If I is terminal node z return u(z)
    If Player(I) = chance
        Return Σ_{a∈A_I} π^C_a · CValue(ha, π₋₁π^C_a, π₁)
    If Player(I) = 2
        Return Σ_{a∈A_I} y_a · CValue(ha, π₋₁y_a, π₁)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += π₋₁ · CValue(ha, π₋₁, π₁x_a)
        Set q(I) = π₁
        Return Σ_{a∈A_I} x_a · CValue(ha, π₋₁, π₁x_a)


CValue(∅, 1)
```

This is the product of the probabilities of prior actions of player $P1$ before arriving at infoset I

Note. Due to perfect recall this product is the same every time we visit the infoset; irrespective of which node of the infoset we arrived at.

The Typical CRM Algorithm Implementation

```
CValue(ActionHistory h, AccOtherProb π_{-1}, AccProb π_1)
    Let I be infoset corresponding to h
    If I is terminal node z return u(z)
    If Player(I) = chance
        Return Σ_{a∈A_I} π_a^C · CValue(ha, π_{-1}π_a^C, π_1)
    If Player(I) = 2
        Return Σ_{a∈A_I} y_a · CValue(ha, π_{-1}y_a, π_1)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += π_{-1} · CValue(ha, π_{-1}, π_1 x_a)
        Set q(I) = π_1
        Return Σ_{a∈A_I} x_a · CValue(ha, π_{-1}, π_1 x_a)


CValue(∅, 1)
```

The Overall Equilibrium Algorithm with CRM

After each period $t \in \{1, \ldots, T\}$:

- With last period behavior strategies x_t, y_t call $\mathrm{CValue}(\emptyset, 1, 1)$
- Store $\tilde{u}_{t,a}$ and $q_t(I)$ for each action a and infoset I of P1
- Symmetrically, do so for player P2
- Update strategies at all information sets

$$\forall j \in \mathcal{J}_1:\ x_{t+1}^j \leftarrow \mathrm{Update}\left(\tilde{u}_t^j\right), \qquad \forall j \in \mathcal{J}_2: y_{t+1}^j \leftarrow \mathrm{Update}\left(\tilde{u}_t^j\right)$$

At the end:

$$\forall I \in \mathcal{J}_1 \forall a \in A_I: x_a^* = \frac{\sum_t q_t(I) x_{t,a}}{\sum_t q_t(I)}$$

$$\forall I \in \mathcal{J}_2 \forall a \in A_I: y_a^* = \frac{\sum_t q_t(I) y_{t,a}}{\sum_t q_t(I)}$$

Approximate Equilibrium in Behavioral Strategies

What algorithm to use for local regret updates?

The Overall Equilibrium A

Any no-regret algorithm for the n-action no-regret problem can be used, e.g. FTRL, OFTRL, EXP, etc.

What performs well in practice is what is known as **Regret Matching**!

After each period $t \in \{1, \ldots, T\}$:

- With last period behavior strategies x_t, y_t call $\mathrm{CValue}(\emptyset, 1, 1)$
- Store $\tilde{u}_{t,a}$ and $q_t(I)$ for each action a and infoset I of P1
- Symmetrically, do so for player P2
- Update strategies at all information sets

$$\forall j \in \mathcal{J}_1: x_{t+1}^j \leftarrow \mathrm{Update}\left(\tilde{u}_t^j\right), \qquad \forall j \in \mathcal{J}_2: y_{t+1}^j \leftarrow \mathrm{Update}\left(\tilde{u}_t^j\right)$$

At the end:

$$\forall I \in \mathcal{J}_1 \forall a \in A_I: x_a^* = \frac{\sum_t q_t(I) x_{t,a}}{\sum_t q_t(I)}$$

$$\forall I \in \mathcal{J}_2 \forall a \in A_I: y_a^* = \frac{\sum_t q_t(I) y_{t,a}}{\sum_t q_t(I)}$$

Approximate Equilibrium in Behavioral Strategies

Regret Matching and Regret Matching+

- Consider the n action no-regret learning setting; at each period we choose $x_t \in \Delta(n)$, observe utility vector u_t and get utility $\langle x_t, u_t \rangle$
- At each period t calculate regret of not playing action a

$$r_{t,a} = u_{t,a} - \langle u_t, x_t \rangle$$

- Calculate cumulative regret of not playing action a

$$R_{t,a} = \sum_{\tau \leq t} r_{t,a} = R_{t-1,a} + r_{t,a}$$

- Choose next distribution, proportional to positive part of regret

$$x_{t+1,a} \propto \left[R_{t,a}\right]^+ := \max\left\{R_{t,a}, 0\right\}$$

- People typically refer to CFR with RegretMatching as simply "CFR"

Regret Matching+

- Consider the n action no-regret learning setting; at each period we choose $x_t \in \Delta(n)$, observe utility vector u_t and get utility $\langle x_t, u_t \rangle$
- At each period t calculate regret of not playing action a
$$r_{t,a} = u_{t,a} - \langle u_t, x_t \rangle$$
- Continuously clip above zero, as you accumulate regret of a
$$R_{t,a} = \left[R_{t-1,a} + r_{t,a}\right]^+$$
- Choose next distribution, proportional to $R_{t,a}$
$$x_{t+1,a} \propto R_{t,a}$$
- Regret Matching and Regret Macthing+ achieve Regret $\leq \sqrt{n/T}$

Extra Tricks for Empirical Improvement

Monte-Carlo Stochastic Approximation of Utilities

- Sample chance move (e.g. sampled A)
- Go to **Infoset 1**

$$\hat{\tilde{u}}_f = -1, \qquad \hat{\tilde{u}}_r = 0$$

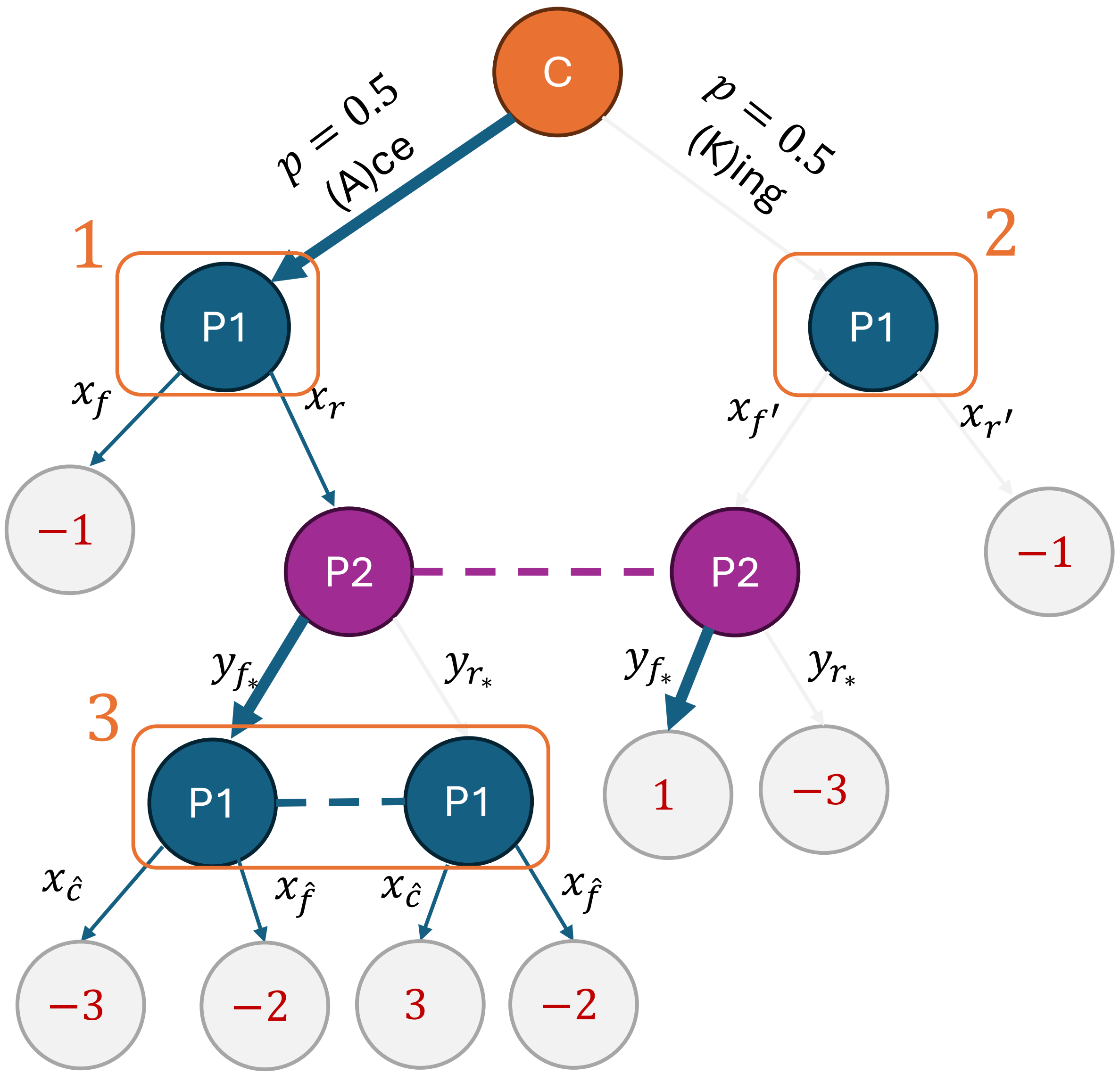
- Go down tree the r path
- Sample P2 move (e.g. sampled f_*)
- Go down to **Infoset 3**

$$\hat{\tilde{u}}_{\hat{c}} = -3, \qquad \hat{\tilde{u}}_{\hat{f}} = -1$$

$$\hat{\tilde{u}}_r \mathrel{+}= x_{\hat{c}}\hat{\tilde{u}}_{\hat{c}} + x_{\hat{f}}\hat{\tilde{u}}_{\hat{f}}$$

- Update probabilities of visited infosets

$$(x_f, x_r) \leftarrow \mathrm{Update}(\hat{\tilde{u}}_f, \hat{\tilde{u}}_r)$$
$$(x_{\hat{c}}, x_{\hat{f}}) \leftarrow \mathrm{Update}(\hat{\tilde{u}}_{\hat{c}}, \hat{\tilde{u}}_{\hat{f}})$$



Typical Monte Carlo Algorithm Implementation

```
MCCValue(ActionHistory h, AccProb π_1)
    Let I be infoset corresponding to h
    If I is terminal node z return u(z)
    If Player(I) = chance
        Sample a ~ π^C
        Return MCCValue(ha, π_1)
    If Player(I) = 2
        Sample a ~ y^I
        Return MCCValue(ha, π_1)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += MCCValue(ha, π_1 · x_a)
        Set q(I) = π_1
        Return Σ_{a∈A_I} x_a · MCCValue(ha, π_1 · x_a)
Value(∅, 1)
```

Can Combine with Update Step in One Pass

```
MCCValue(ActionHistory h, AccProb π_1)
    Let I be infoset corresponding to h
    If I is terminal node z return u(z)
    If Player(I) = chance
        Sample a ~ π^C
        Return MCCValue(ha, π_1)
    If Player(I) = 2
        Sample a ~ y^I
        Return MCCValue(ha, π_1)
    If Player(I) = 1
        For a ∈ A_I: ũ_a += MCCValue(ha, π_1 · x_a)
        Set q(I) = π_1
        Update x^I_next ← Update(ũ^I)
        Return Σ_{a∈A_I} x_a · MCCValue(ha, π_1 · x_a)
```

Alternation

After each period t:

- If t is odd then update the strategy of the x-player
- If t is even then update strategy of the y-player

For most natural algorithms, alternation can only help in terms of reducing the violation of best response constraints!

Can converge faster to equilibrium

Weighted Averaging

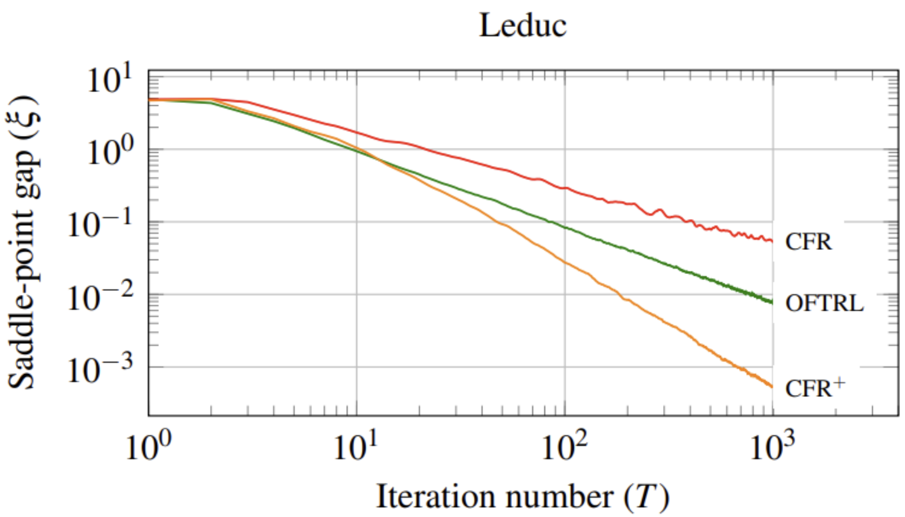
- Instead of uniformly weighting all rounds, put more weight on more recent rounds of play

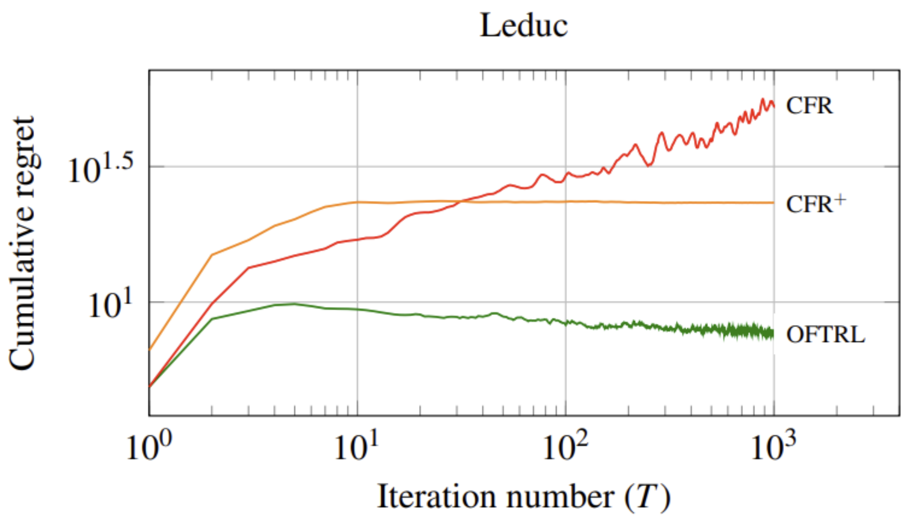
$$\frac{1}{\sum_t t^\alpha} \sum_t t^\alpha \tilde{x}_t$$

- Typically, one uses linear averaging (i.e., $\alpha = 1$)

- The CFR algorithm that uses RegretMatching+, alternation and linear averaging is typically referred to as “CFR+”

Empirical Comparisons





Violations of best response

saddle-point gap

$$\boxed{\mathrm{Regret}_y(x_*, y_*) + \mathrm{Regret}_x(x_*, y_*)} := \max_y x_*^\top A y - x_*^\top A y_* + x_*^\top A y_* - \min_x x^\top A y_* = \boxed{\max_y x_*^\top A y - \min_x x^\top A y_*}$$

$$\boxed{R_y + R_x} = \max_y \bar{x}^\top A y - \frac{1}{T}\sum_t x_t^\top A y_t + \frac{1}{T}\sum_t x_t^\top A y_t - \min_x x^\top A \bar{y} = \boxed{\max_y \bar{x}^\top A y - \min_x x^\top A \bar{y}}$$

Sum of learning algorithm regrets

saddle-point gap of average strategies $\bar{x}, \bar{y}$

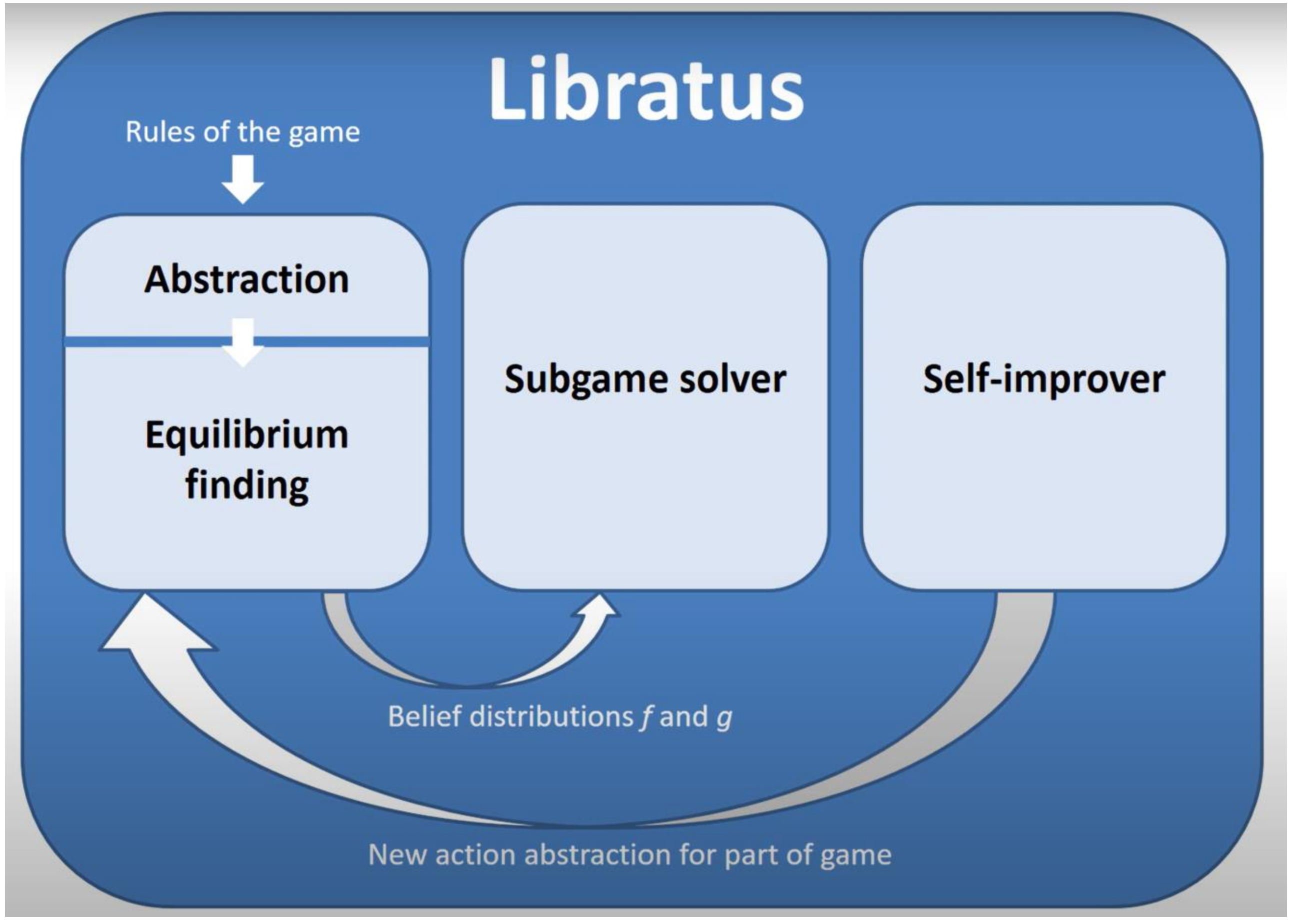
Elements of the Libratus AI

- The first agent to achieve superhuman performance in two player No-Limit Texas Hold’em poker (10^{161} decision points)

- Prior best was Limit Texas Hold’em (10^{13} decision points); solution is basically “run CFR+”

- For No-Limit Texas Hold’em game is too big for this approach!

Elements of Libratus AI



Credits: Superhuman AI for heads-up no-limit poker: Libratus beats top professionals (youtube.com)