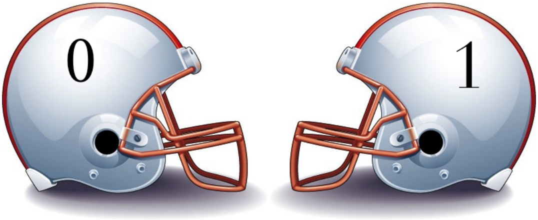
Efficient Probabilistically Checkable Debates

Andrew Drucker
MIT

Polynomial-time Debates

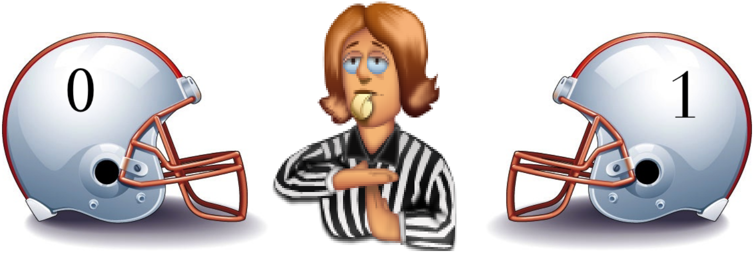


- Given: language L, string x;
- Player 1 argues that $x \in L$; Player 0 argues $x \notin L$.
- k-round debate:

$$y = (y^1, y^2, \ldots, y^k)$$

- $y^i = i^{th}$ move; P1 plays odd-numbered moves; $\left|y^i\right| \leq \mathrm{poly}(|x|)$.

Polynomial-time Debates



- Polynomial-time verifier: Boolean function $V(x, y)$
- V is a **debate system for** L if

$$x \in L \iff \text{P1 wins under optimal play (can force } V(x, y) = 1\text{)}$$

Polynomial-time Debates

Theorem (Chandra, Stockmeyer '76)

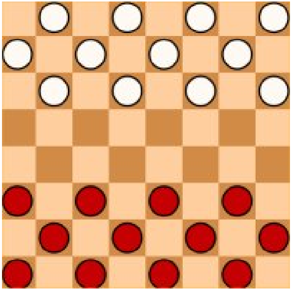
L has a $\mathrm{poly}(n)$-round, polynomial-time debate system

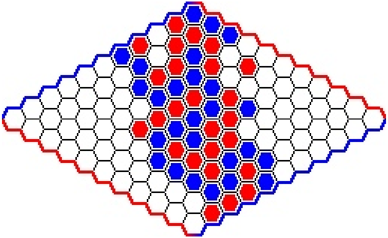
$$\iff \quad L \in \mathrm{PSPACE}.$$

- Debate characterization of PSPACE lets us prove many natural problems are PSPACE-complete!

Applications

- E.g., n-by-n Checkers, Hex, many other 2-player games are PSPACE-complete:





Probabilistic Verifiers

- What happens if we restrict the form of the debate verifier?
- Say that a debate system is **probabilistically checkable** if $V(x, y)$ inspects only $O(1)$ bits of the debate string y

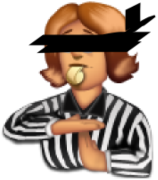
  (and decides debate with perfect completeness and $1/3$ soundness, say).

Probabilistic Verifiers

INPUT: x

Probabilistic Verifiers

INPUT: x

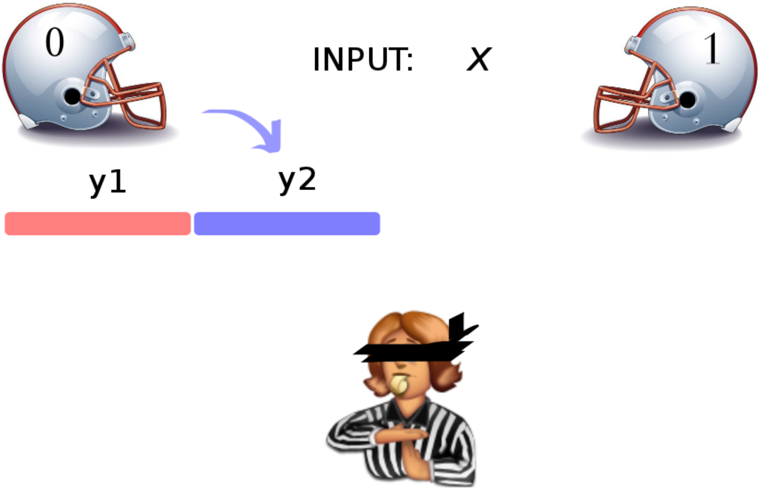


Probabilistic Verifiers

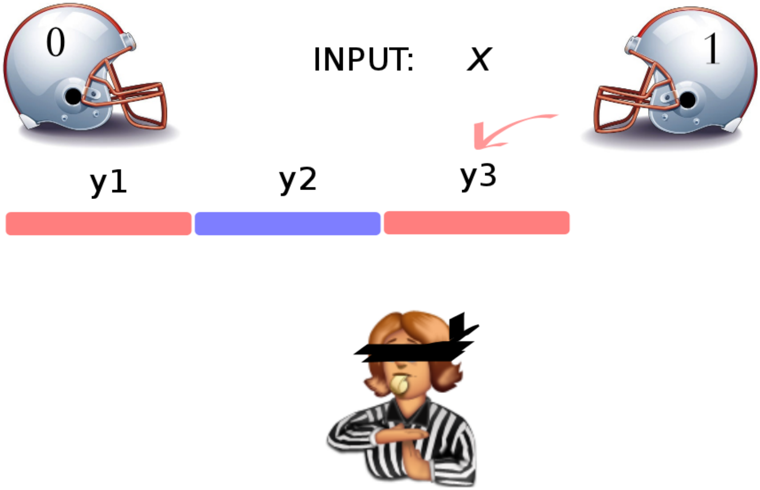
INPUT: x

y1

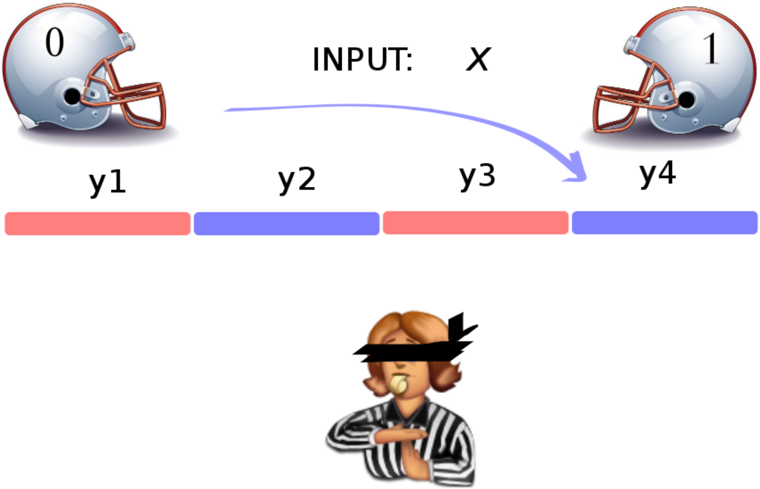
Probabilistic Verifiers



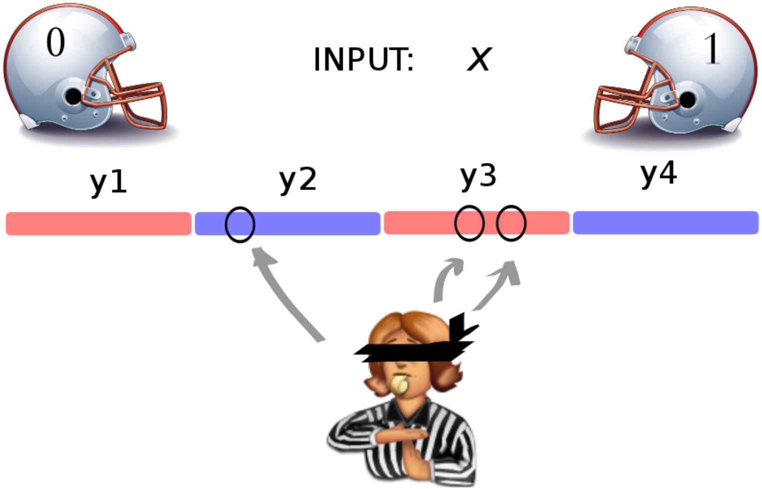
Probabilistic Verifiers



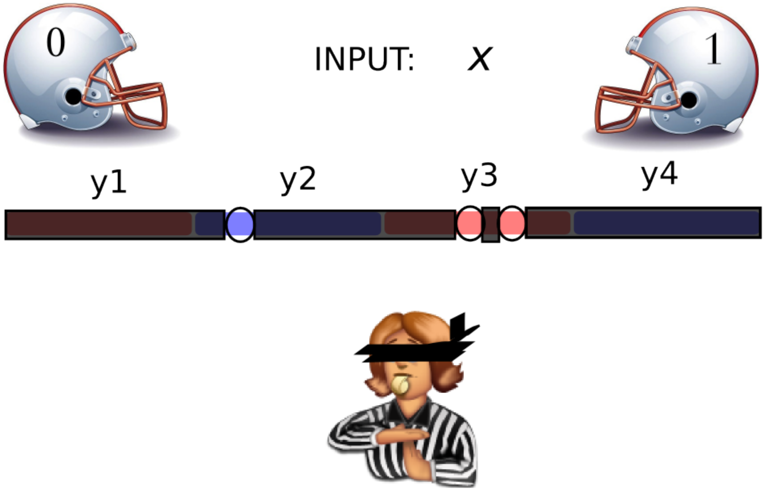
Probabilistic Verifiers



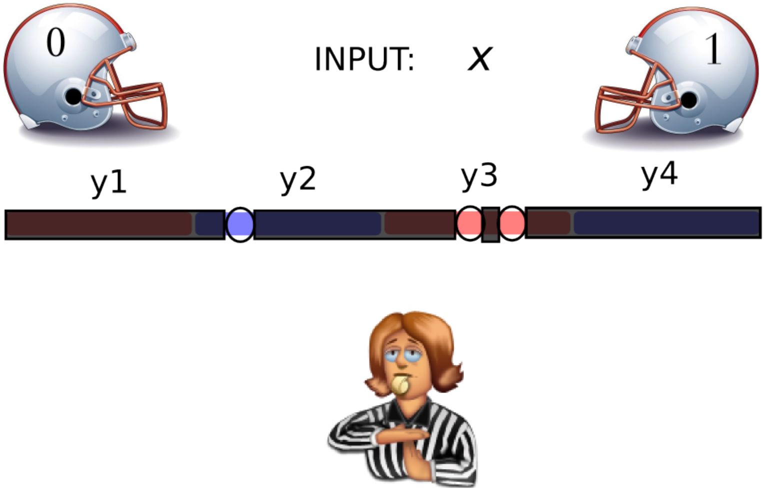
Probabilistic Verifiers



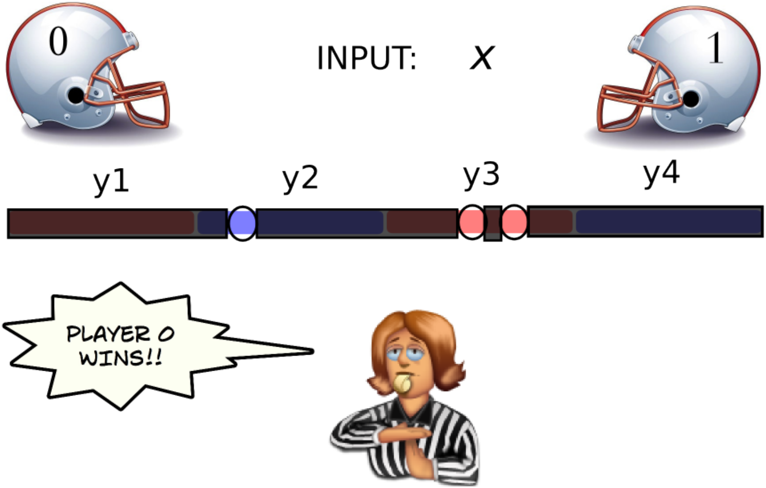
Probabilistic Verifiers



Probabilistic Verifiers



Probabilistic Verifiers



Probabilistic Verifiers

Theorem (Condon, Feigenbaum, Lund, Shor '95)

$L \in \mathsf{PSPACE} \Leftrightarrow$

L has a $\mathrm{poly}(n)$-round, **probabilistically checkable** *debate system (**PCDS**),*

with a verifier using $O(\log n)$ bits of randomness.

("PCP Characterization of PSPACE")

PCP Characterizations of Complexity Classes

- Analogous PCP characterizations were shown for:
  1. Polynomial Hierarchy **[Ko, Lin '94]**;
  2. IP = PSPACE **[CFLS '97]**;
  3. AM **[D. '11]**.

Our result

We strengthen **[CFLS]**:

Theorem

Suppose $L \in \mathrm{PSPACE}$ *has a poly-time debate system defined by uniform circuits of size* $s = s(n)$.

Then, L *has a PCDS with a debate of total bitlength* $\widetilde{O}(s)$,

whose verifier uses $\log s + \log(\mathrm{polylog}(s))$ *bits of randomness.*

Applications

- Like the PCP Theorem, the PCDS Theorem of **[CFLS]** has implications for *hardness of approximation.*
- (For PSPACE-hardness, naturally!)

A natural PSPACE-complete problem

- Input: a 3-CNF formula

$$\psi(z_1, \ldots, z_t)$$

- **Game:** Players take turns assigning values to $z_1, z_2, \ldots$
- P1 wants to *maximize* fraction of satisfied clauses;
  P0 wants to *minimize*.
- Let

$$\mathsf{Val}(\psi)$$

$=$ (fraction of satisfied clauses of ψ under optimal play).

A natural PSPACE-complete problem

- PSPACE-complete to compute Val(ψ) exactly.
- **[CFLS]** implies: PSPACE-complete to compute Val(ψ) to within a suff. small additive error $\varepsilon > 0$.

Application

- Improved parameters $\longrightarrow$ better conditional hardness results!
- Suppose computing $\mathsf{Val}(\psi)$ *exactly* requires $T(n) = n^{\omega(1)}$ time on length-n inputs (infinitely often).
- Then, **[CFLS]** $\Rightarrow$ computing $\mathsf{Val}(\psi) \pm \varepsilon$ requires time $T(n^{\alpha})$, for some $\alpha < 1$.
- Our improvement implies: computing $\mathsf{Val}(\psi) \pm \varepsilon$ requires $T(n/\operatorname{polylog} n)$ time.

Our debate system

- A brief sketch of our construction...
- **Main Step:** Efficiently transform an ordinary debate system for $L \in \mathsf{PSPACE}$ into one that is "stable."

Stable debate systems

- Given: an ordinary debate system $V(x; y^1, \ldots, y^k)$ for L.
- Say that V is **stable** if:

  for all $x \notin L$, Player 0 can force $y = (y^1, \ldots, y^k)$ to be $\Omega(1)$-far in relative distance from any y' for which $V(x; y') = 1$.

Stable debate systems

- How to turn ordinary debates into stable ones?
- **Our tool:** new application of *error-resilient communication protocols*.

Error-resilient communication

- Analogue of error-correcting encoding for 2-way communication **[Schulman '93]**
- Alice and Bob want to hold a chatroom conversation, of a total length T bits.
- Unreliable channel: adversary can corrupt a δ fraction of the transmitted bits (adaptively).

Error-resilient communication

Theorem (Schulman, '93 — Informal)

There is a protocol to simulate T-bit conversations, that uses $T' = O(T)$ bits of communication and succeeds against up to $T'/240$ corrupted bits.

- **[Braverman, Rao '11]**: new protocol $\mathcal{P}_{BR}$ with better parameters: tolerates nearly $1/8$ fraction of errors—and, simpler!
- Both protocols make inspired use of special codes called *tree codes*.

Terminology

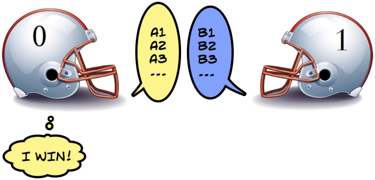
- **perfect execution** of $\mathcal{P}_{BR}$: no transmission errors occur
- else, **noisy execution**

Our application

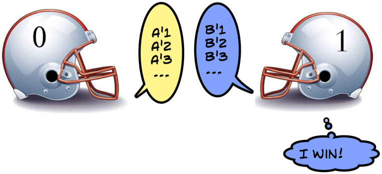
- Let V be an ordinary debate system for L, definable by size-$s(n)$ circuits.
- In V', suppose we can "force" players to encode their moves in V using a perfect execution of $\mathcal{P}_{BR}$. Then:
- **Claim:** V' is stable!
- **Proof:** enough to show: perfect executions with distinct outcomes are well-separated in Hamming distance.

Proof idea

Suppose this perfect execution

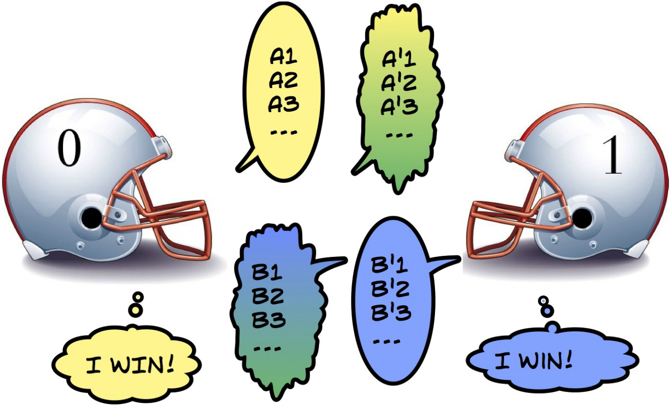


is $T'/10$-close in Hamming distance to this one:



Proof idea

- Then, this noisy execution



has $\leq T'/10$ transmission errors, and causes $\mathcal{P}_{BR}$ to fail.

Can't happen!

Forcing compliance

- So, V' is stable.—Technicality: stable *for perfect executions...*
- How to make the debaters follow $\mathcal{P}_{BR}$?
- (Need to do so *efficiently*.)

Forcing compliance

Lemma

There is an $O(1)$-round debate system D_{BR}, definable by uniform circuits of size $O(T)$, to decide whether a communication transcript w is a valid perfect execution of $\mathcal{P}_{BR}[T]$.

- Use D_{BR} as a "sub-debate" to make our overall debate stable.
- Property that $O(1)$ rounds are used is important:
  can easily make D_{BR} itself stable (using error-correcting codes).

Forcing compliance

Lemma

There is an $O(1)$-round debate system D_{BR}, definable by uniform circuits of size $O(T)$, to decide whether a communication transcript w is a valid perfect execution of $\mathcal{P}_{BR}[T]$.

- Proving the lemma—our main technical challenge:
  1. No *explicit* examples of tree codes known! (Debaters have to "guess and check" a code to use.)
  2. No efficient *decoder* known for any tree code.
  3. Most significantly, the use of tree codes in the Braverman-Rao protocol is somewhat complex, and our efficiency requirements are severe.

Stable $\rightarrow$ prob. checkable

- **Final step:** convert our stable debate system into a probabilistically checkable one.
- **Key tool:** *PCPs of Proximity (PCPPs)* **[Ben-Sasson, Goldreich, Harsha, Sudan, Vadhan '04; Dinur, Reingold '04]**.
- Powerful variant of PCPs; we use an efficient construction from **[Dinur '07]**.

Stable $\rightarrow$ prob. checkable

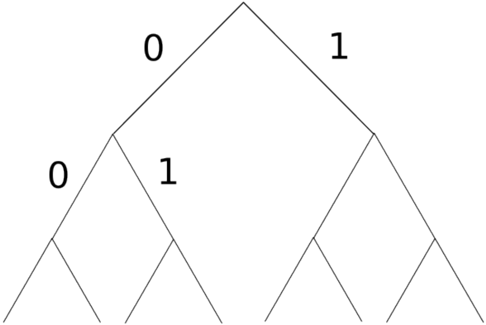
- Basic idea:
  1. Run our stable debate for L;
  2. Ask Player 1 to "certify" his victory, using a PCPP.
- PCPP-like objects also used in **[CFLS]** (in a different way).

More on error-resilient communication

- A small peek...
- First, what are "conversations" exactly?

What are "conversations?"

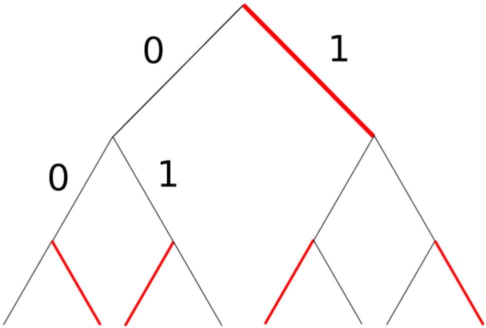
Setting: binary tree of depth T



What are "conversations?"

Setting: binary tree of depth T

Alice's input: X, a degree-1 subset of *odd-depth* edges.

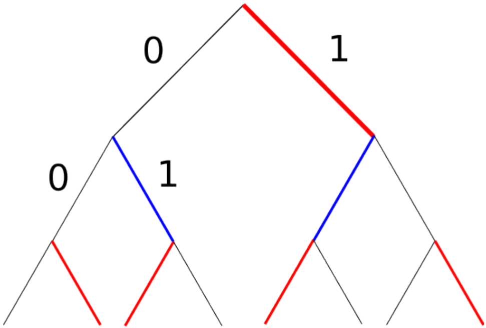


What are “conversations?”

Setting: binary tree of depth T

Alice’s input: X, a degree-1 subset of *odd-depth* edges.

Bob’s input: Y, a degree-1 subset of *even-depth* edges



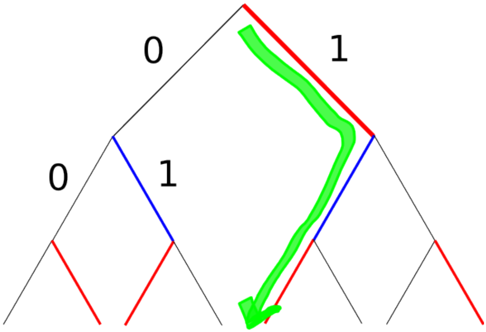
What are "conversations?"

Setting: binary tree of depth T

Alice's input: X, a degree-1 subset of *odd-depth* edges.

Bob's input: Y, a degree-1 subset of *even-depth* edges

Output: the path P determined by X, Y

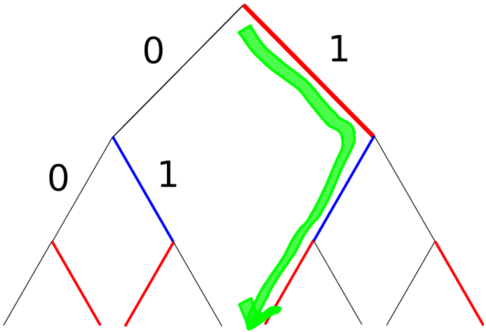


What are "conversations?"

Our application:

X, Y = P1, P0 *strategies* in V

P = resulting debate string



What are "conversations?"

- Also known as the Pointer Jumping problem (PJ_T).
- **[Schulman '93]**: $\exists$ an error-resilient protocol to solve PJ_T using $O(T)$ bits of communication.

Tree codes

- k-**ary tree code** of depth d:

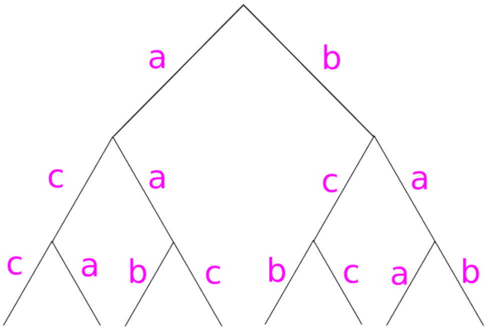
$$C : [k]^{\leq d} \longrightarrow \Sigma$$

- Labeling of edges of the complete k-ary tree of depth d.

Example

Here $k = 2, d = 3$,

$\Sigma = \{a, b, c\}$.



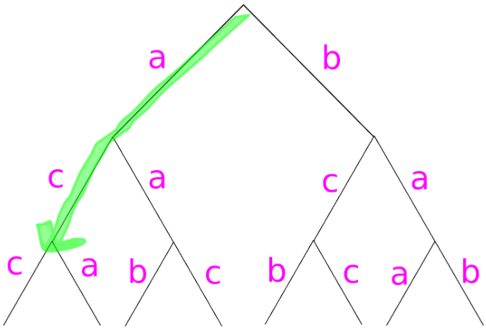
Example

For a path P, define

$$\overline{C}(P)$$

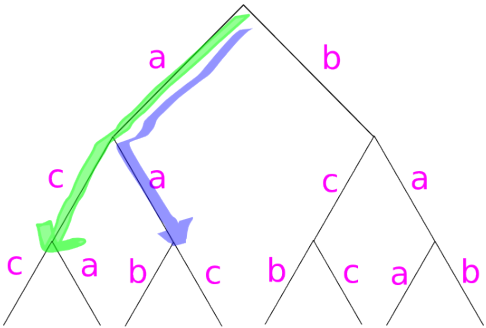
as the concatenation of labels along P.

E.g., $\overline{C}(0,0) = (a, c)$



Example

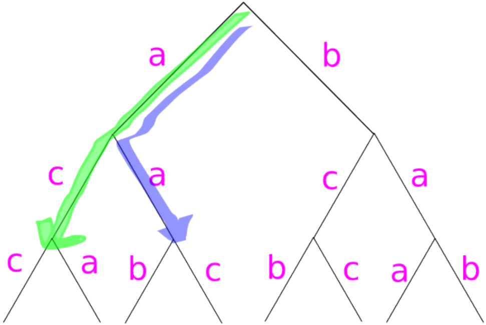
Note: if P, P' agree for t steps, so will $\overline{C}(P)$ and $\overline{C}(P')$.



The distance property

Say that C is a tree code of distance $\alpha \in [0, 1]$, if:

For all pairs P, P' of equal length, $\overline{C}(P)$ and $\overline{C}(P')$ differ on at least an α fraction of places where they *could* differ!



The distance property

- Braverman-Rao protocol for PJ_T requires 5-ary tree codes of depth $d = \Theta(T)$, distance $\alpha = \Omega(1)$, alphabet size $|\Sigma| = O(1)$.
  (Schulman: similar.)
- These exist, but no *explicit* construction is known.
- (**Explicit** $\leftrightarrow$ $C(\cdot)$ computable in time $\mathsf{poly}(T)$.)
- Schulman gave a probabilistic construction using $O(T)$ bits of randomness—good enough for our application!

Schulman's tree codes

- Fix k (arity of tree); let $p = O_k(1) \gg k$ be a prime.
- **The random seed:** $r = (r_1, \ldots, r_d) \in \mathbb{F}_p^d \quad (d =$ depth).
- **The tree code:**

$$C_{(r)}(x_1, \ldots, x_t) := \sum_{j=1}^{t} x_j \cdot r_{t+1-j}.$$

- Has distance $\Omega(1)$ w.h.p.!

An open question

- Debates where P0 plays *randomly* also characterize PSPACE **[Shamir '90]**.
- **[CFLS '97]**: these debates can also be made prob. checkable.
- Give a similar efficiency improvement for these debates?

Thanks!