Learning Finite Automata Using Label Queries

LEV REYZIN

YALE UNIVERSITY → YAHOO! RESEARCH

DANA ANGLUIN

Yale University

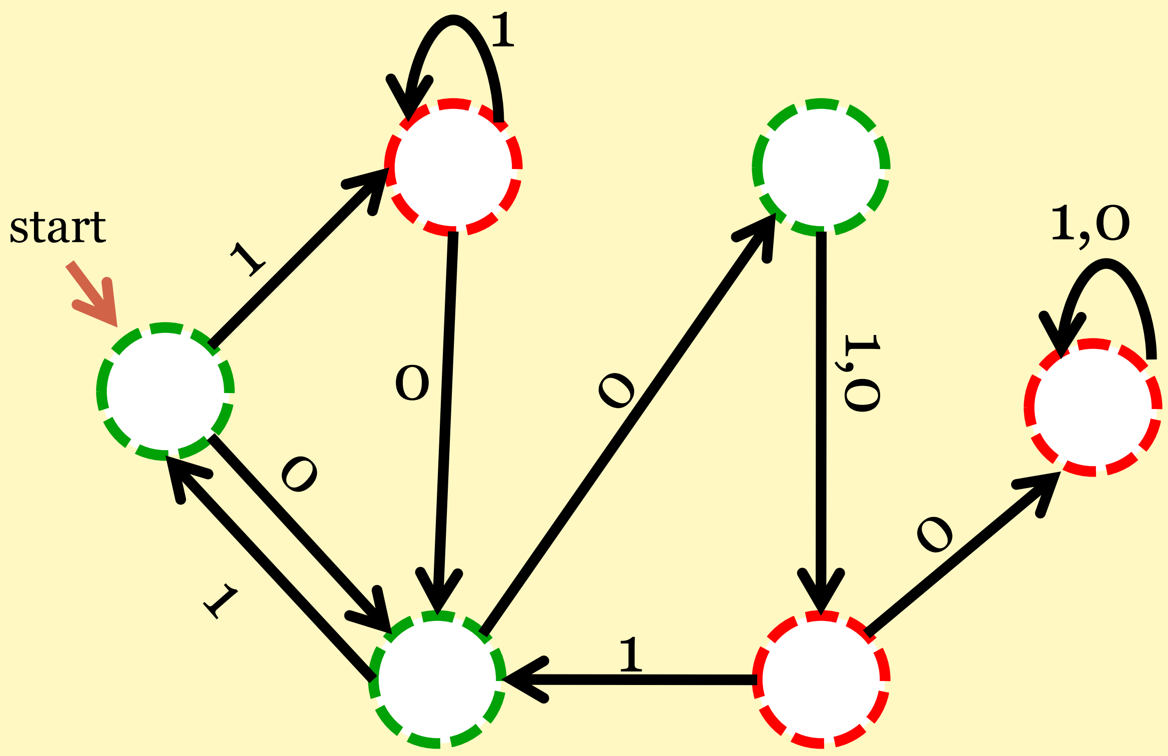
LEONOR BECERRA-BONACHE

Yale University & Rovira i Virgili

ADRIAN HORIA DEDIU

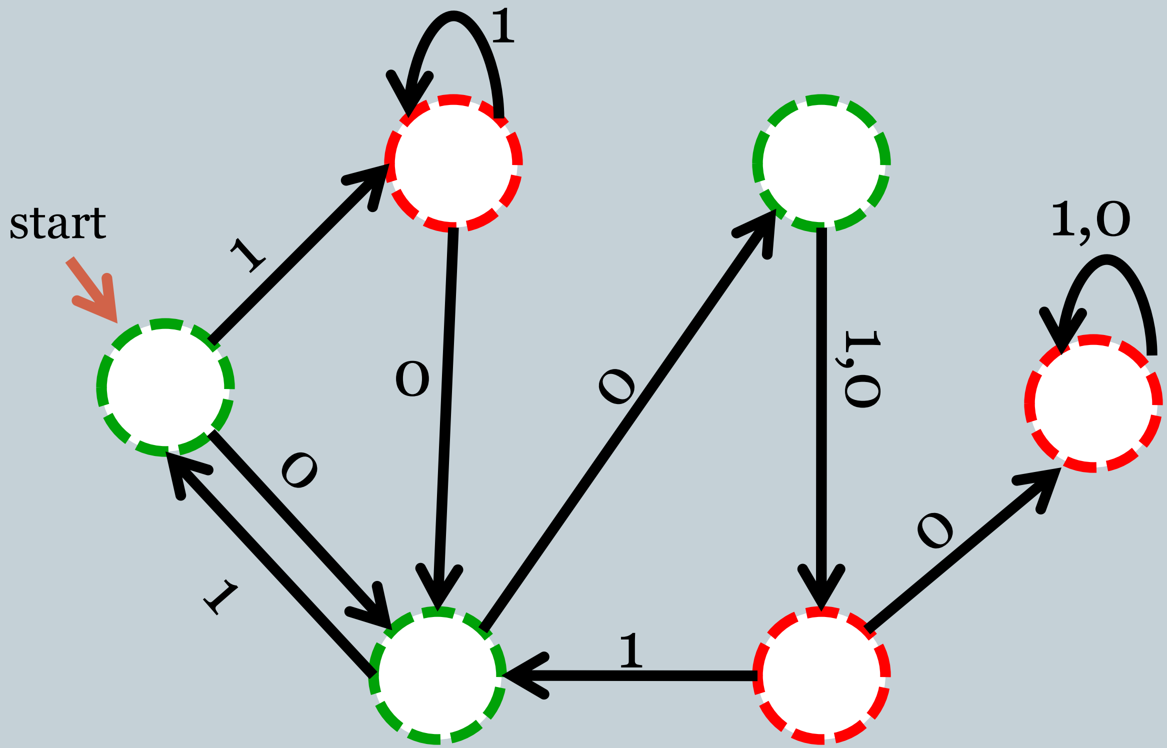
Rovira i Virgili & Politehnica Bucharest

Membership Queries



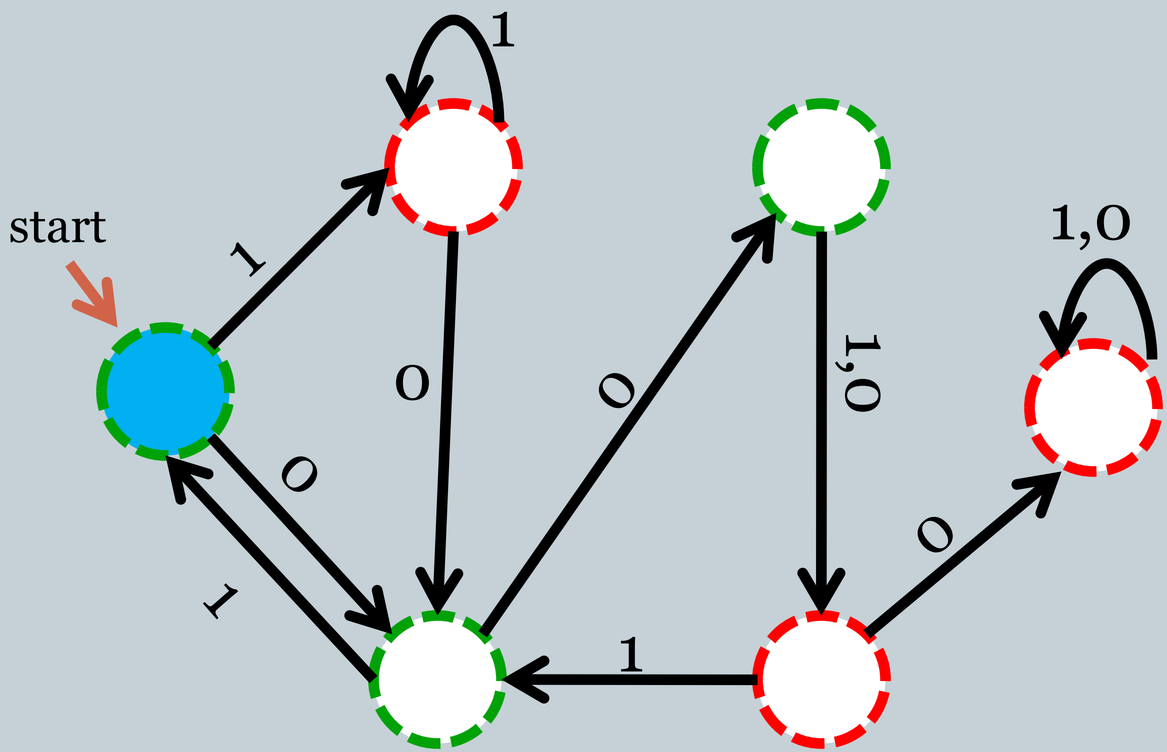
hidden from learner

Membership Queries



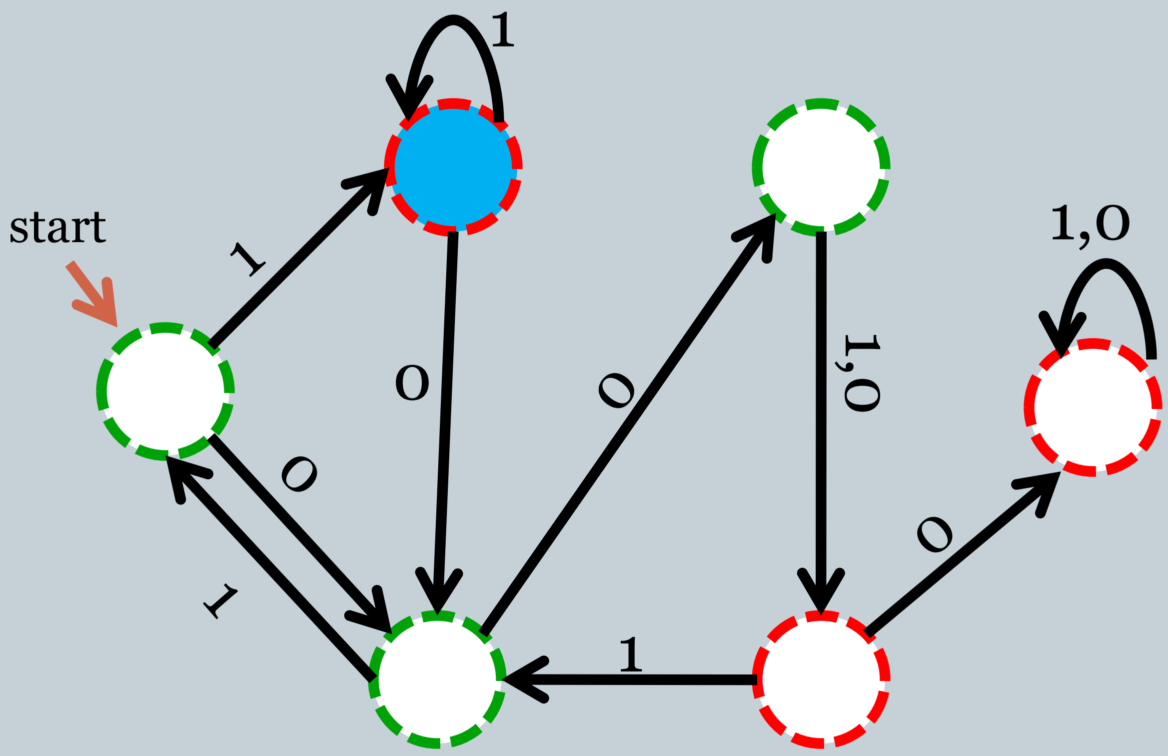
M("10011") = ?

Membership Queries



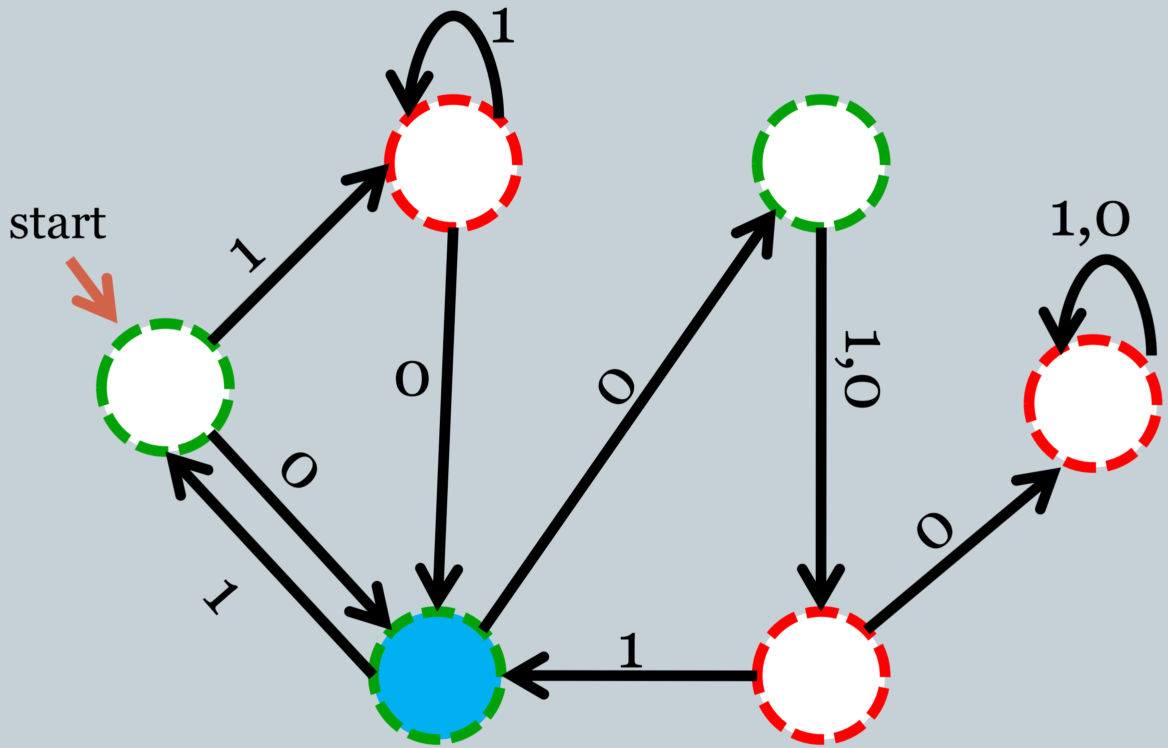
M("10011") = ?

Membership Queries



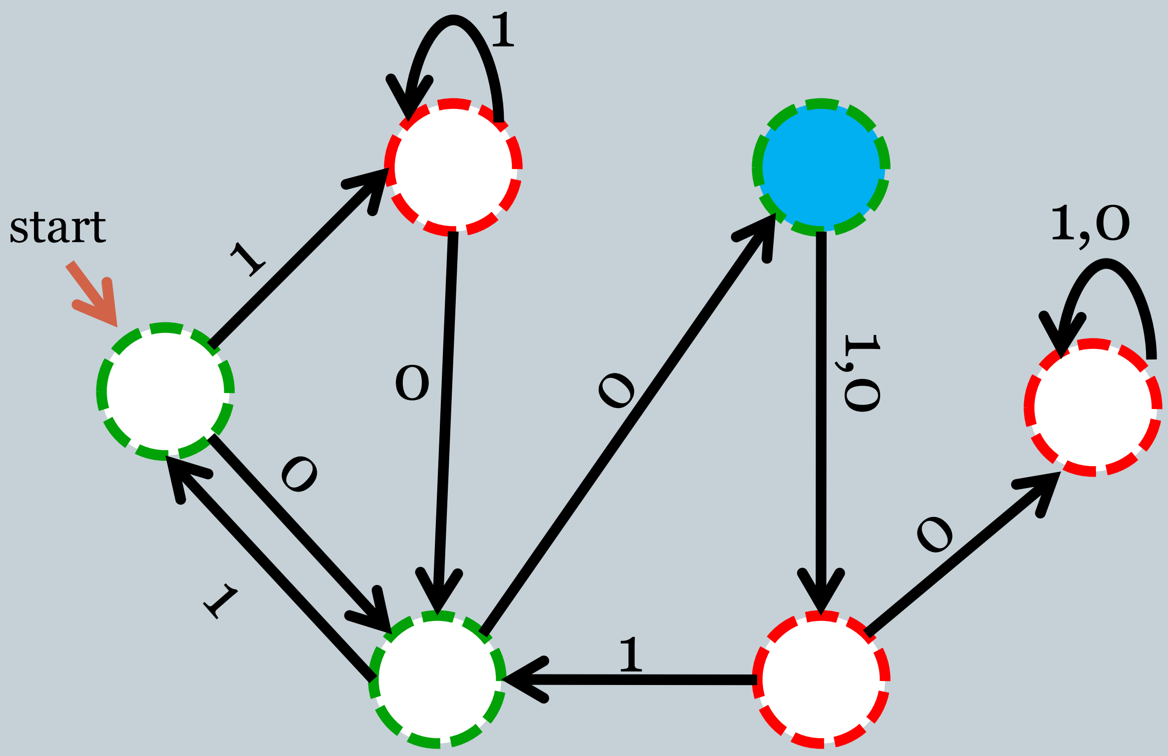
M("10011") = ?

Membership Queries



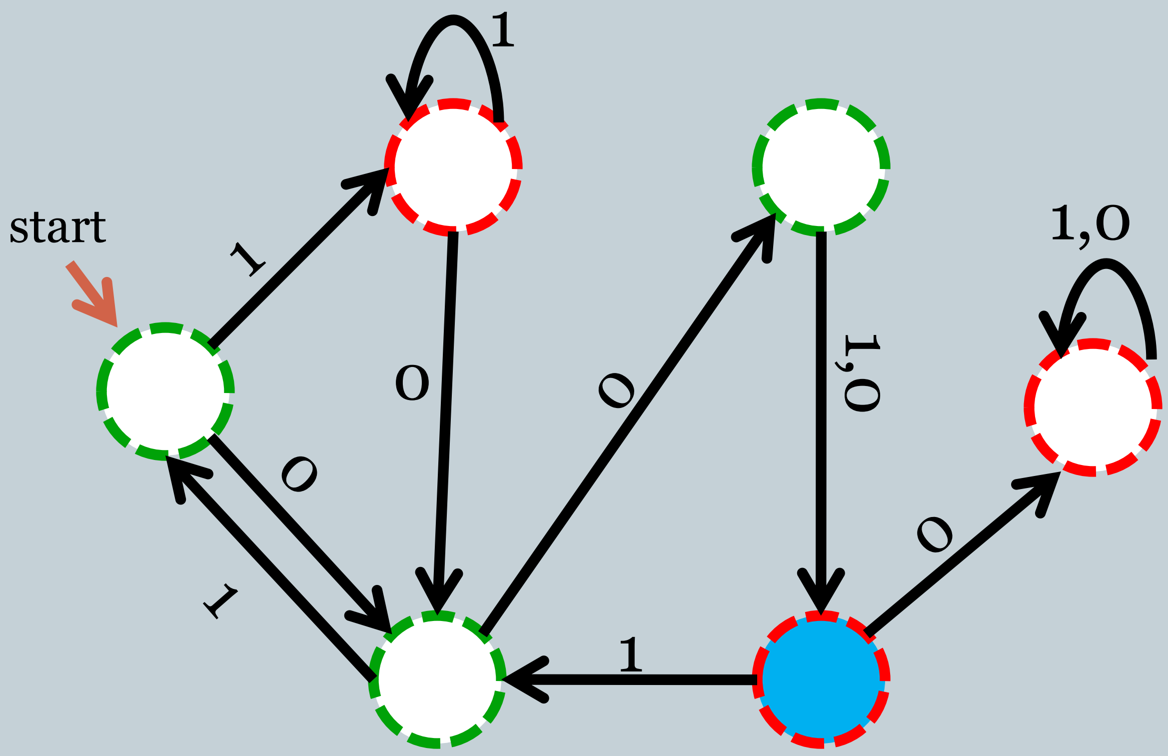
M("10011") = ?

Membership Queries



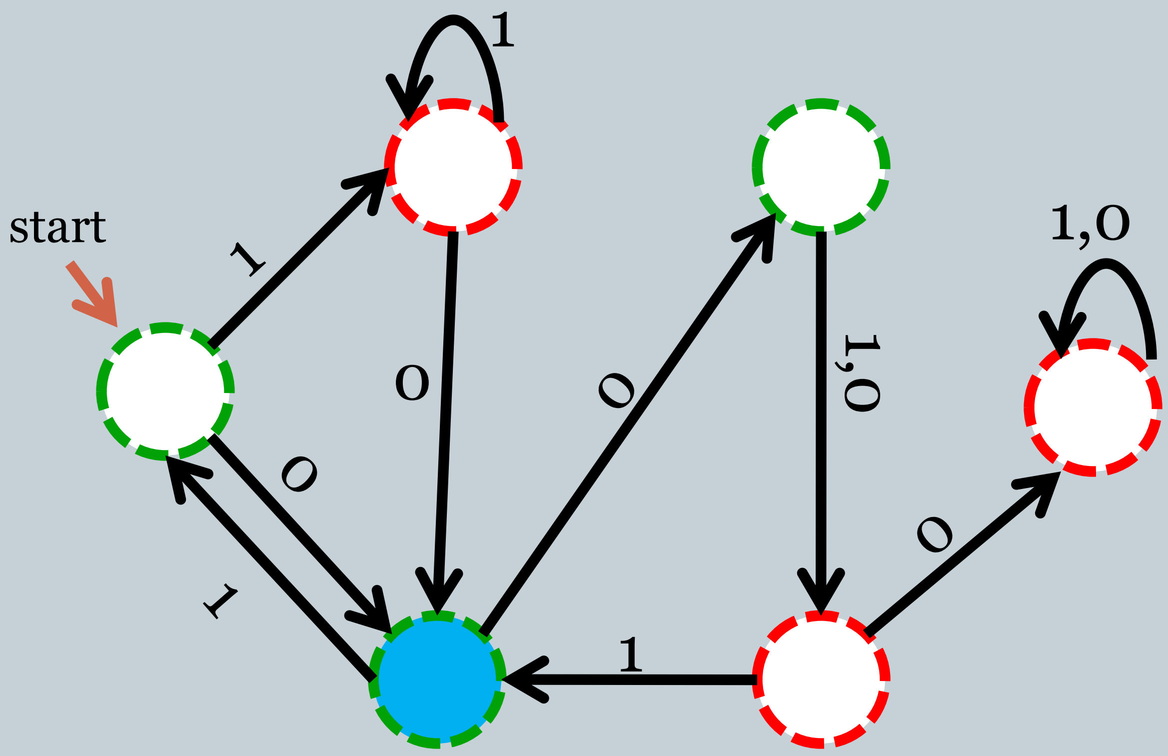
M("10011") = ?

Membership Queries



M(“10011”) = ?

Membership Queries



M(“10011”) = Accept

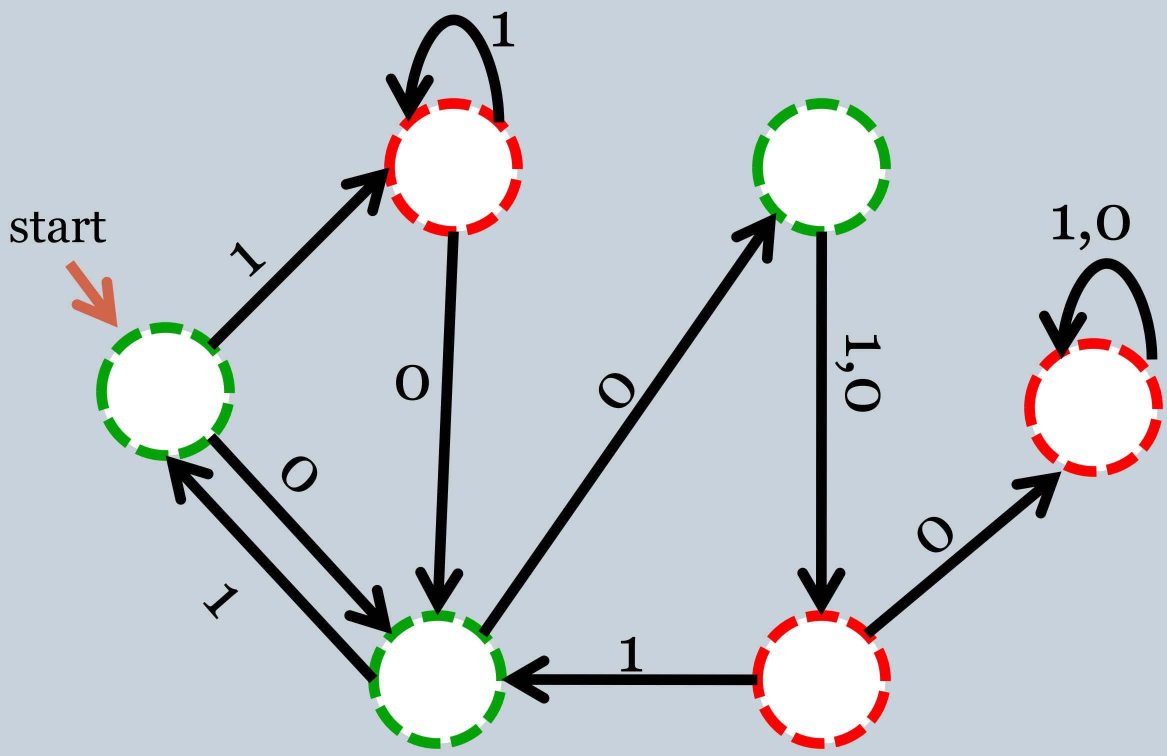
Major Past Work

- [Angluin '87] gave an algorithm for learning DFAs (regular languages) with membership and equivalence queries.
  - impossible (in poly time) with membership queries alone
- [Freund et al. '97] showed how to predict the labels seen on a random walk on a DFA of arbitrary topology but randomly determined accept/reject behavior.
- [Becerra-Bonache et al. '06] introduced correction queries, which are a generalization of MQs.
  - correction queries give extra information when the learner lands on a reject state

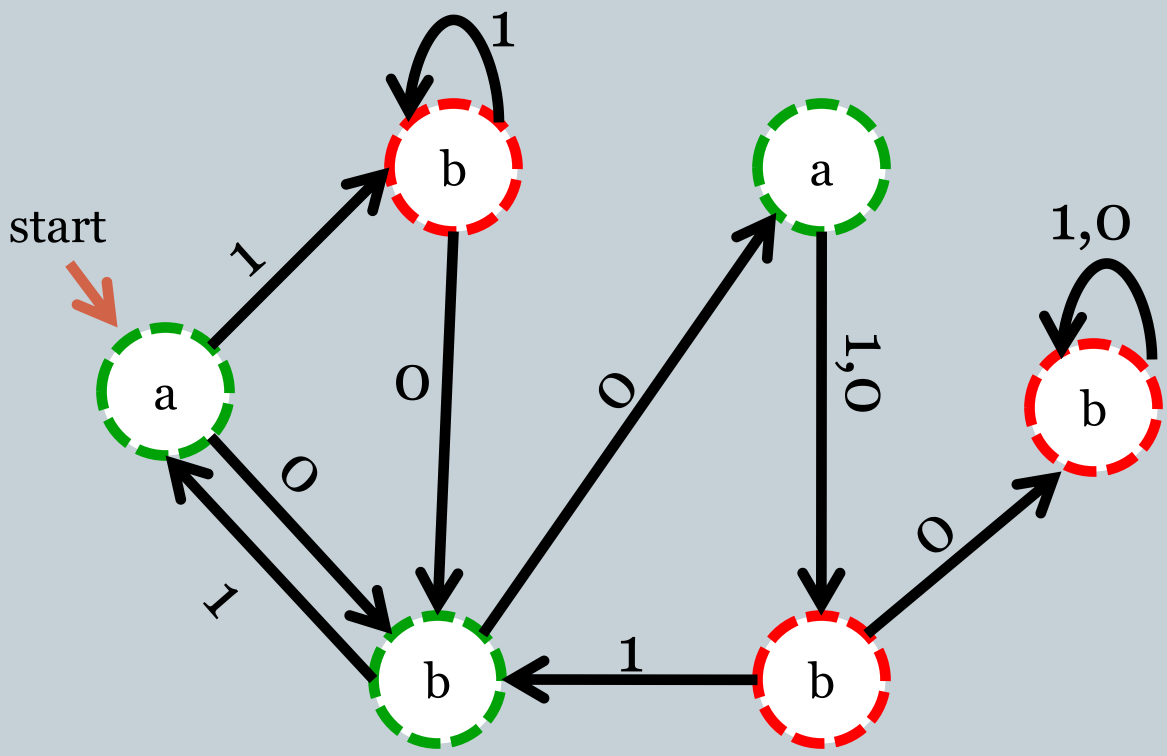
We Introduce Label Queries: the Model

- Given a hidden DFA, M
  - teacher labels states of M using labels from L to make machine M^L
  - learner picks strings s, and asks label queries $M^L(s)$
  - learner must construct a machine H that is output equivalent to M
  - teacher and learner can agree on any strategy, including "coding tricks"
- Label queries generalize correction queries.
  - the teacher can choose what "extra information" is returned and is not limited to "corrections"
- We can analyze the case when the teacher does a random labeling
  - this is similar to the [Freund et al. '97] setting

Learning FSA with Label Queries

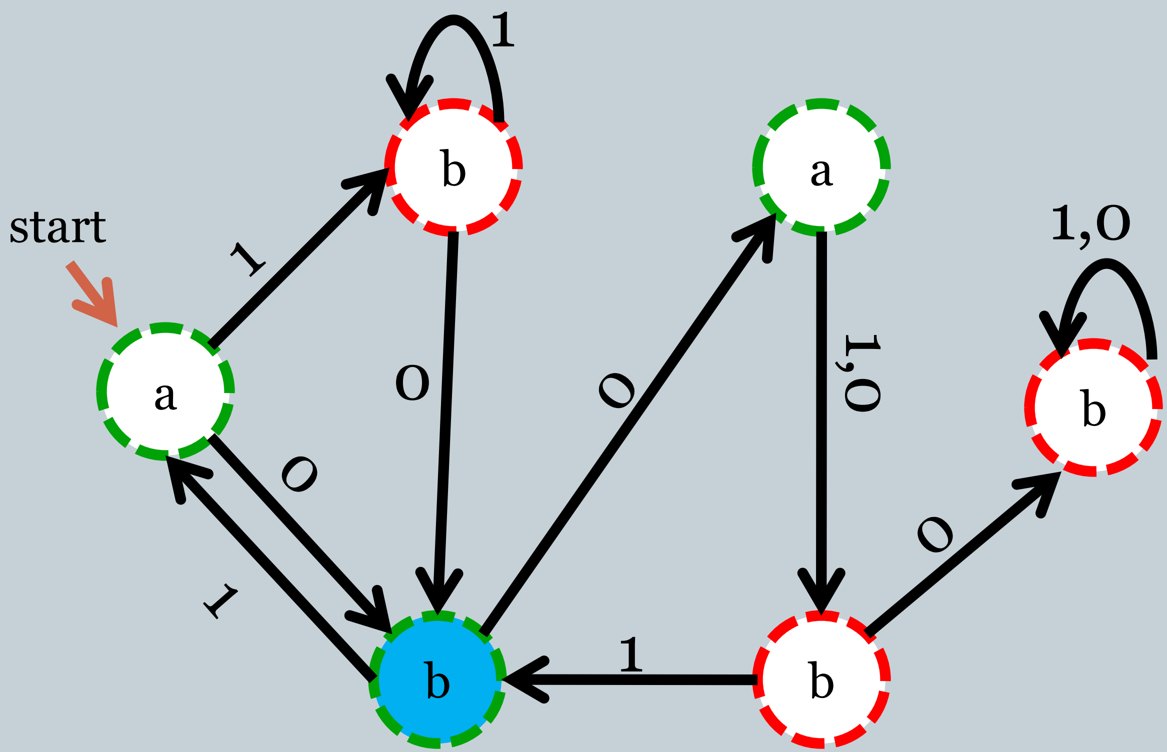


Learning FSA with Label Queries



L = {a,b}

Learning FSA with Label Queries



M^L(“10011”) = (Accept,b)

Results for Helpful Teacher

For a machine on n states, with input alphabet X, and output alphabet Y, and labels from L placed helpfully

- A "helpful" teacher:
  - if teacher uses |L| labels, learner must use $\Omega(|X|\ n \log n / (1+\log |L|))$ queries
  - teacher can use n labels so that learner uses $|X|\ n$ queries
  - teacher can use $2^{|X|}$ labels s.t. learner uses $O(|X|\ n^2)$ queries
  - for c-concentrating automata with in-degree at most k, teacher can use $3k|X|+c$ labels s.t. learner uses $O(|X|\ n \log n)$ queries.

Results for Helpful Teacher

For a machine on n states, with input alphabet X, and output alphabet Y

- A "helpful" teacher:
  - If teacher uses |L| labels, learner must use $\Omega(|X|\ n \log n / (1+\log |L|))$ queries.
    - information theoretic lower bound
  - teacher can use n labels so that learner uses $|X|\ n$ queries
    - trivial
  - teacher can use $2^{|X|}$ labels s.t. learner uses $O(|X|\ n^2)$ queries
  - for c-concentrating automata with in-degree at most k, teacher can use $3k|X|+c$ labels s.t. learner uses $O(|X|\ n \log n)$ queries.

Results for Helpful Teacher

For a machine on n states, with input alphabet X, and output alphabet Y

- A "helpful" teacher:
  - If teacher uses |L| labels, learner must use $\Omega(|X|\, n \log n / (1+\log |L|))$ queries.
    - information theoretic lower bound
  - teacher can use n labels so that learner uses |X| n queries
    - trivial
  - teacher can use $2^{|X|}$ labels s.t. learner uses $O(|X|\, n^2)$ queries
  - for c-concentrating automata with in-degree at most k, teacher can use 3k|X|+c labels s.t. learner uses $O(|X|\, n \log n)$ queries.

Use n Labels s.t. Learner Uses |X|n Queries

- teacher gives each state unique label.
- learner explores all |X| transitions for each newly reached state.
- n states, and |X| transitions to explore per state.
- Can't do better because of information theoretic lower bound.

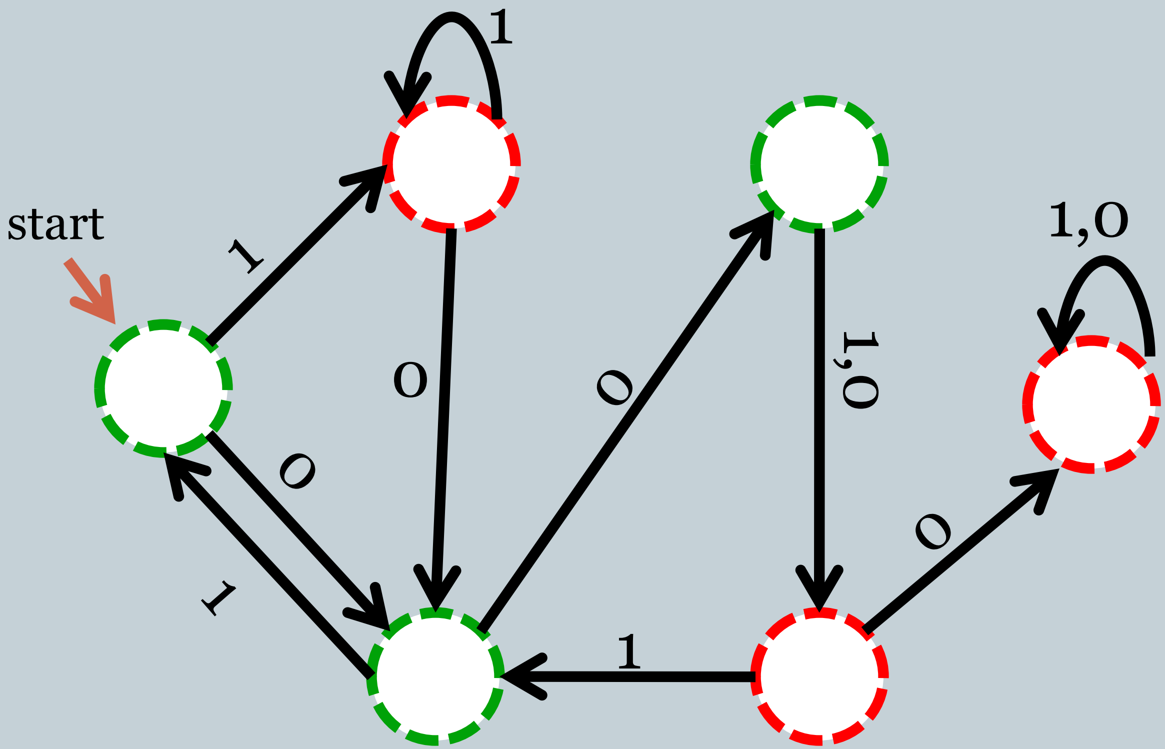
Results for Helpful Teacher

For a machine on n states, with input alphabet X, and output alphabet Y

- A "helpful" teacher:
  - If teacher uses |L| labels, learner must use $\Omega(|X|\ n \log n\ /\ (1+\log |L|))$ queries.
    - information theoretic lower bound
  - teacher can use n labels so that learner uses |X| n queries
    - trivial
  - teacher can use $2^{|X|}$ labels s.t. learner uses $O(|X|\ n^2)$ queries
  - for c-concentrating automata with in-degree at most k, teacher can use 3k|X|+c labels s.t. learner uses $O(|X|\ n \log n)$ queries.

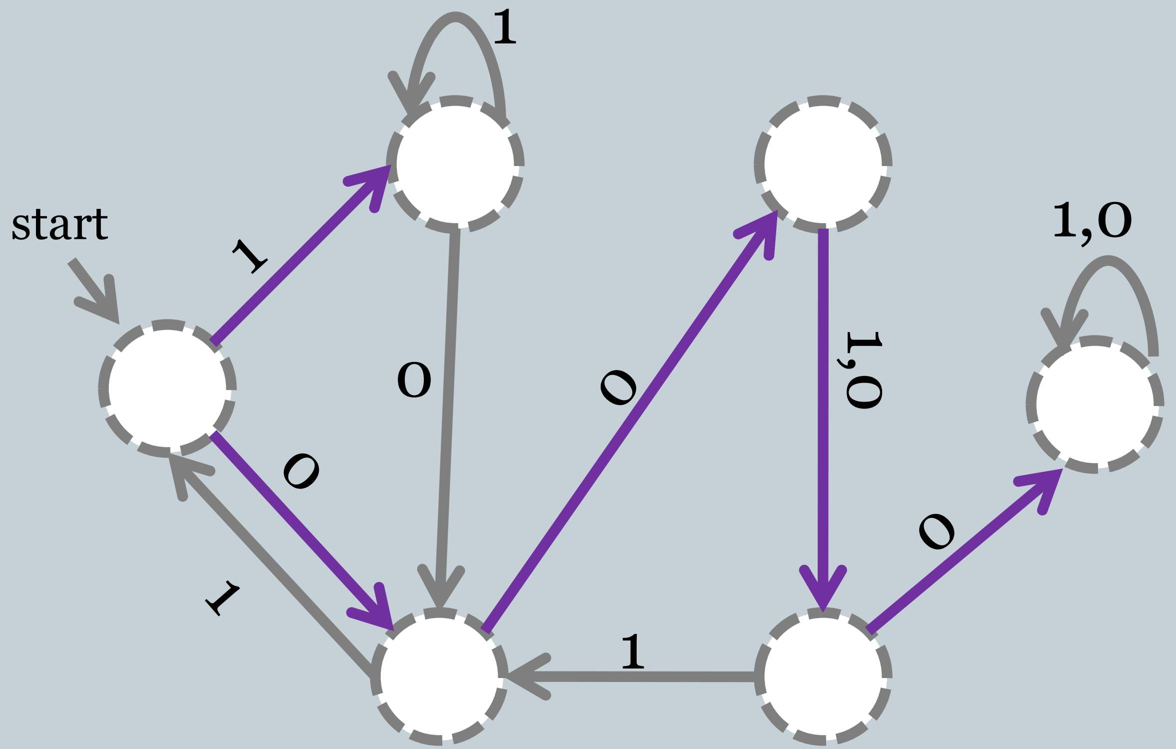
Use $2^{|X|}$ Labels s.t. Learner Uses $O(|X|\ n^2)$ Queries

- For illustration, assume $|X| = 2$



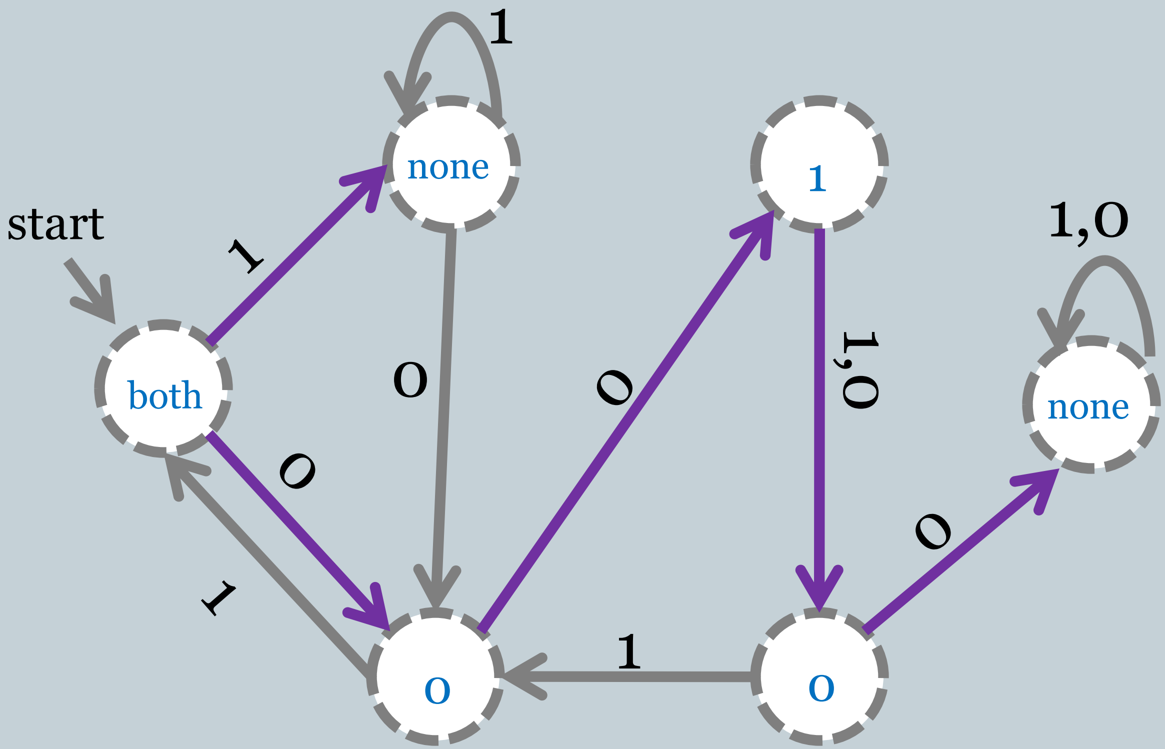
Use $2^{|X|}$ Labels s.t. Learner Uses $O(|X|\, n^2)$ Queries

- For illustration, assume $|X| = 2$



Use $2^{|X|}$ Labels s.t. Learner Uses $O(|X|\, n^2)$ Queries

- For illustration, assume $|X| = 2$
  - uses L = {none, 0, 1, both}, so $|L| = 2^2 = 4$



Use $2^{|X|}$ Labels s.t. Learner Uses $O(|X|\, n^2)$ Queries

- A live complete sample for a language L is a set of strings P, that for every state q (other than the dead state) of the minimal acceptor for L, contains a string that leads from the start state to q.
- Given the spanning tree labeling by the teacher, the learner can construct a live complete sample of size n.
- [Angluin '81] showed that given a live complete sample P, a learner can find the regular language using $O(|X||P|n) = O(|X|n^2)$.

Results for Helpful Teacher

For a machine on n states, with input alphabet X, and output alphabet Y

- A “helpful” teacher:
  - If teacher uses |L| labels, learner must use $\Omega(|X| \, n \log n / (1+\log |L|))$ queries.
    - information theoretic lower bound
  - teacher can use n labels so that learner uses |X| n queries
    - trivial
  - teacher can use $2^{|X|}$ labels s.t. learner uses $O(|X| \, n^2)$ queries
  - for c-concentrating automata with in-degree at most k, teacher can use 3k|X|+c labels s.t. learner uses $O(|X| \, n \log n)$ queries.

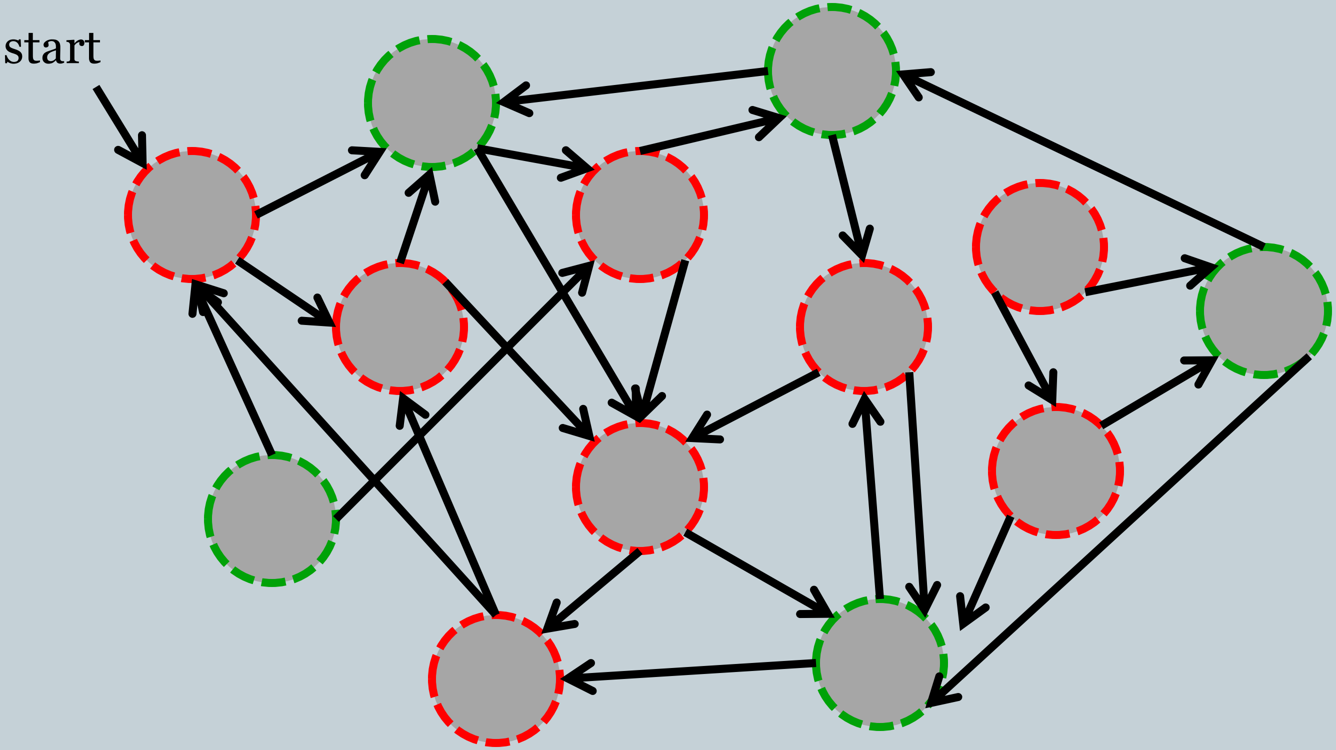
Result for c-concentrating Automata

- An automaton M is c-concentrating if there is some set Q of at most c states of M such that every state of M can reach at least one state in Q.
  - Every strongly connected automaton is 1-concentrating.

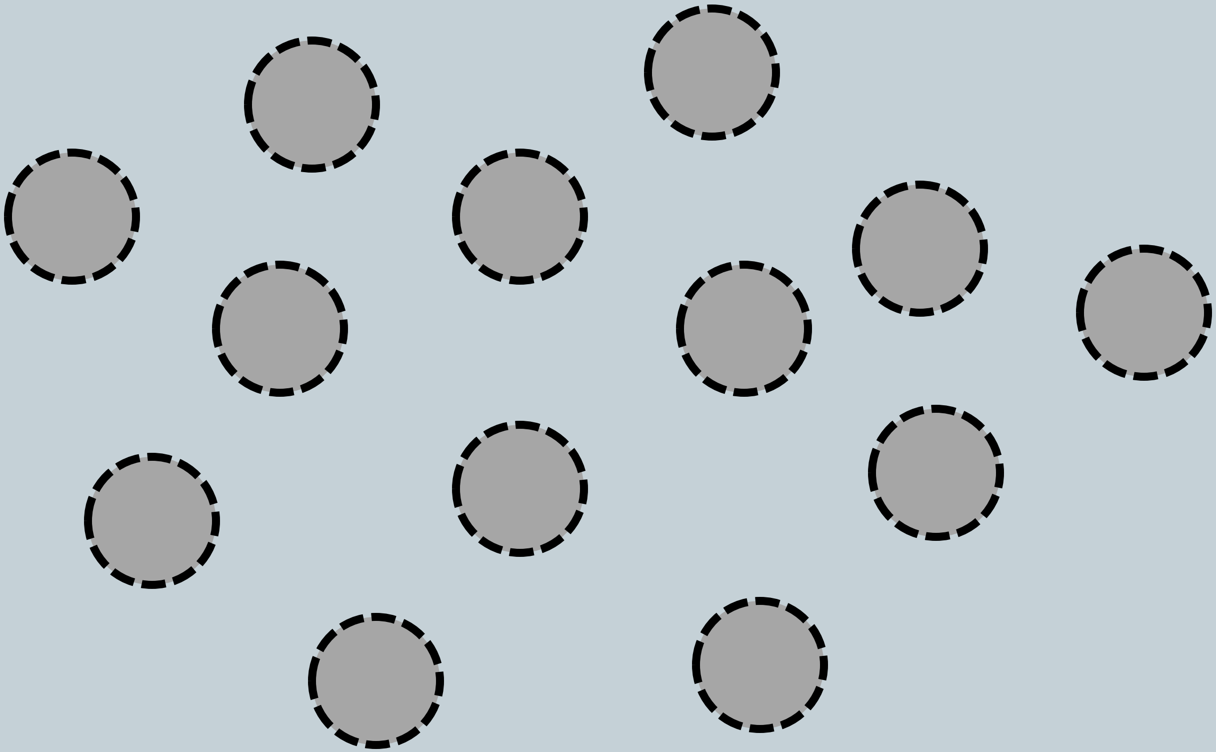
- For c-concentrating automata with in-degree at most k, the teacher can use $3k|X|+c$ labels s.t. learner uses $O(|X|\, n \log n)$ queries.

- We will give the idea for the proof for 1-concentrating automata, with $|X| = 2$.

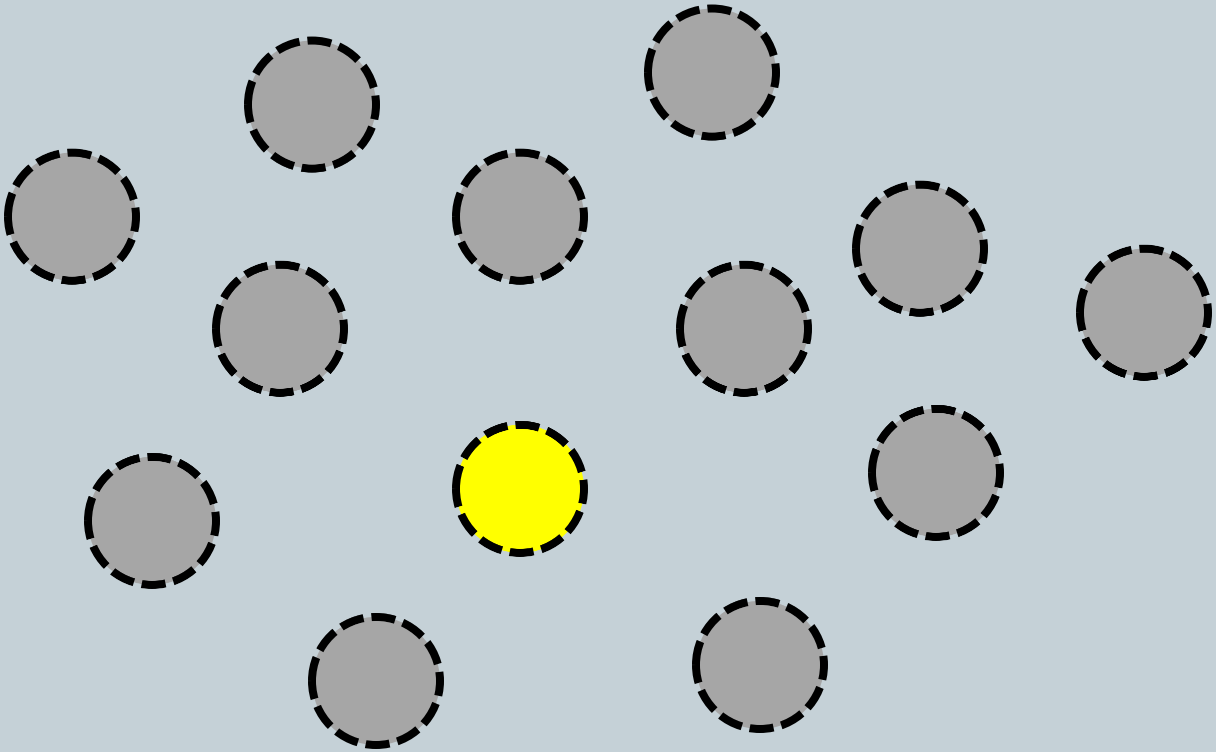
Result for c-concentrating Automata



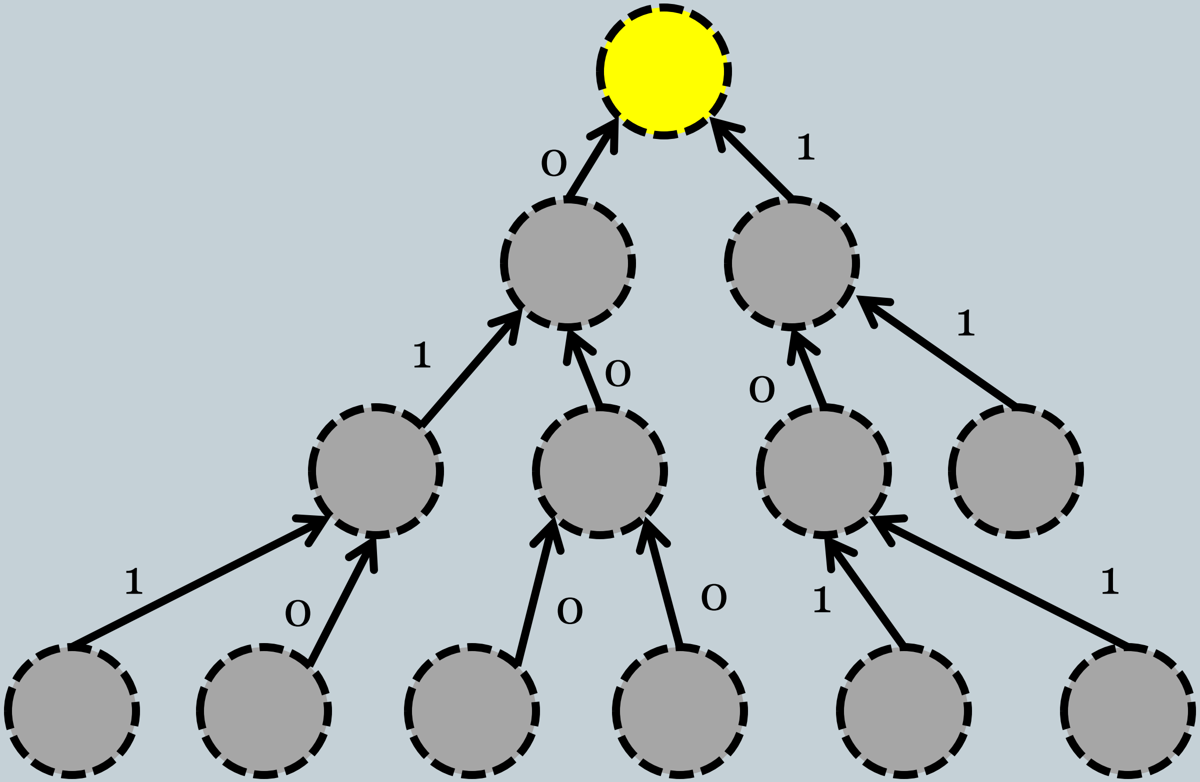
Result for c-concentrating Automata



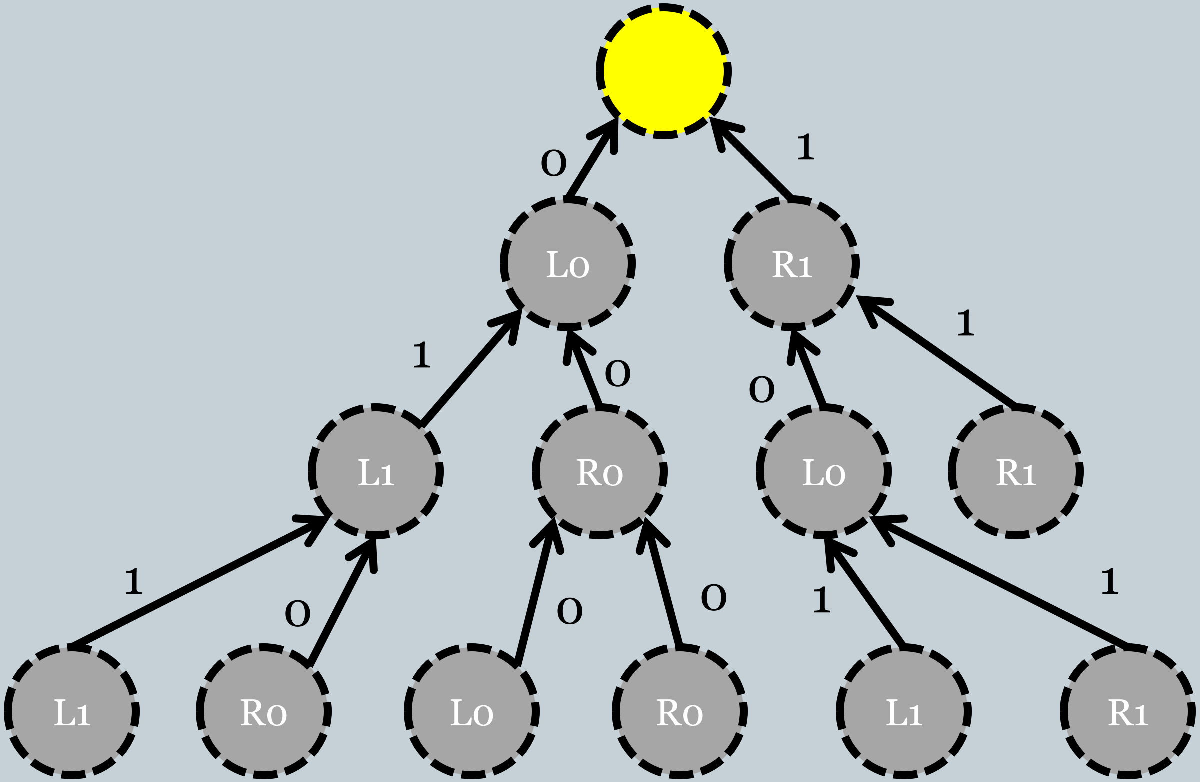
Result for c-concentrating Automata



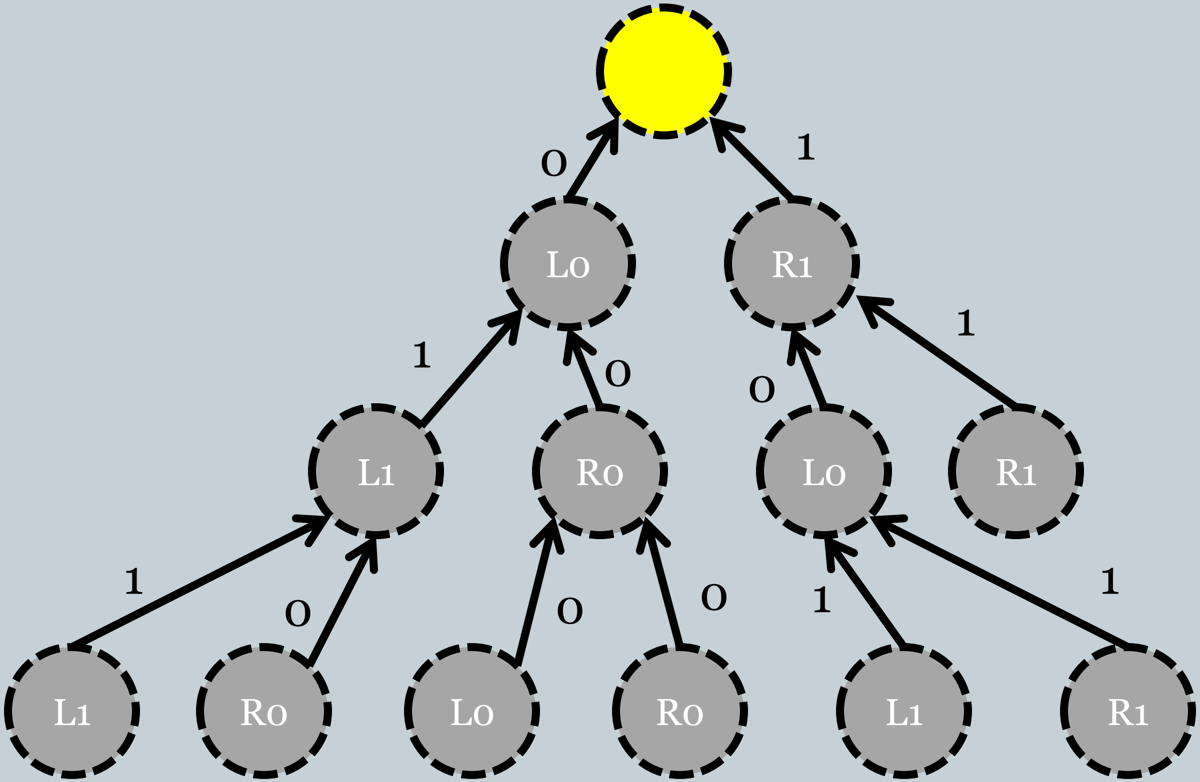
Result for c-concentrating Automata



Result for c-concentrating Automata

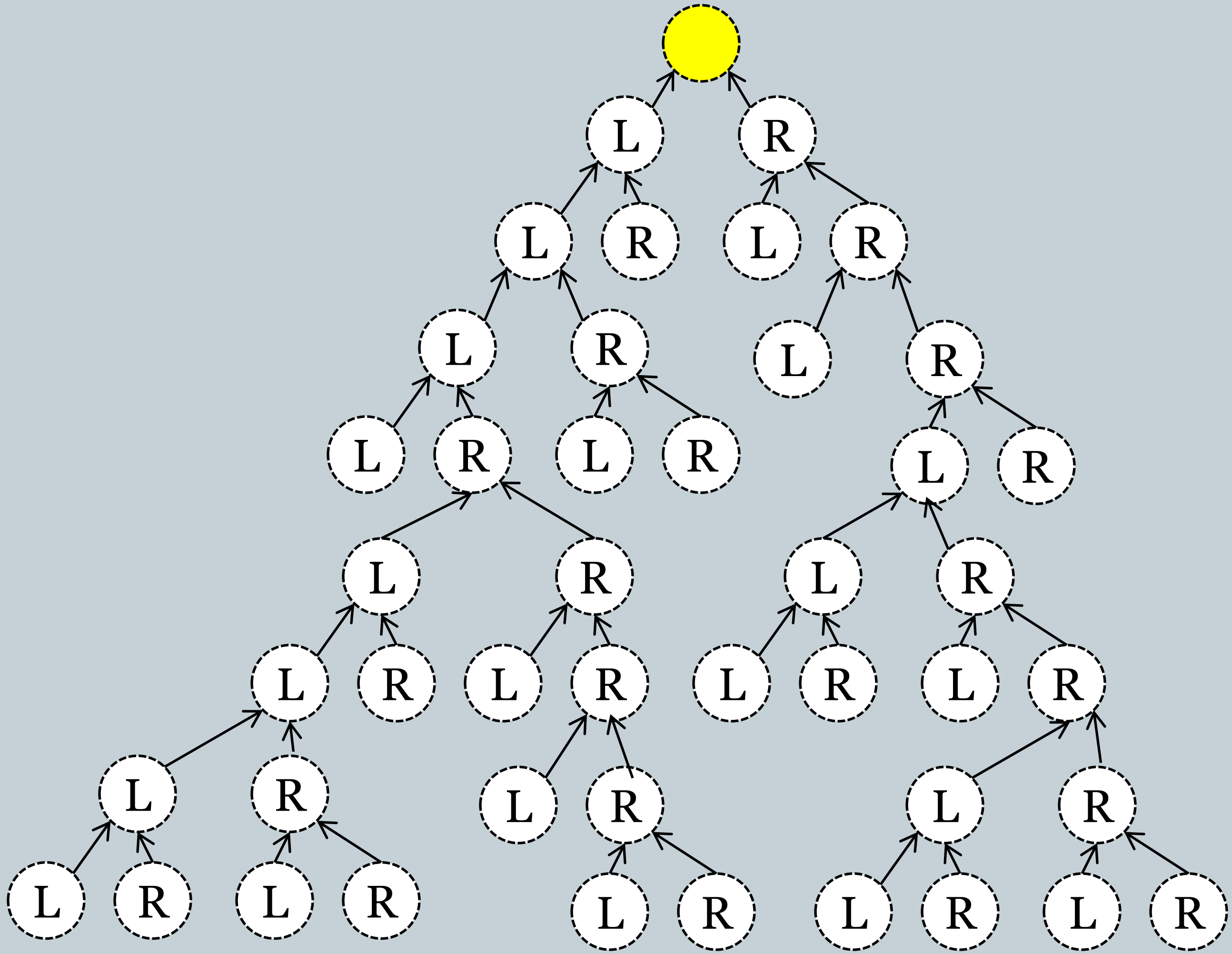


Result for c-concentrating Automata

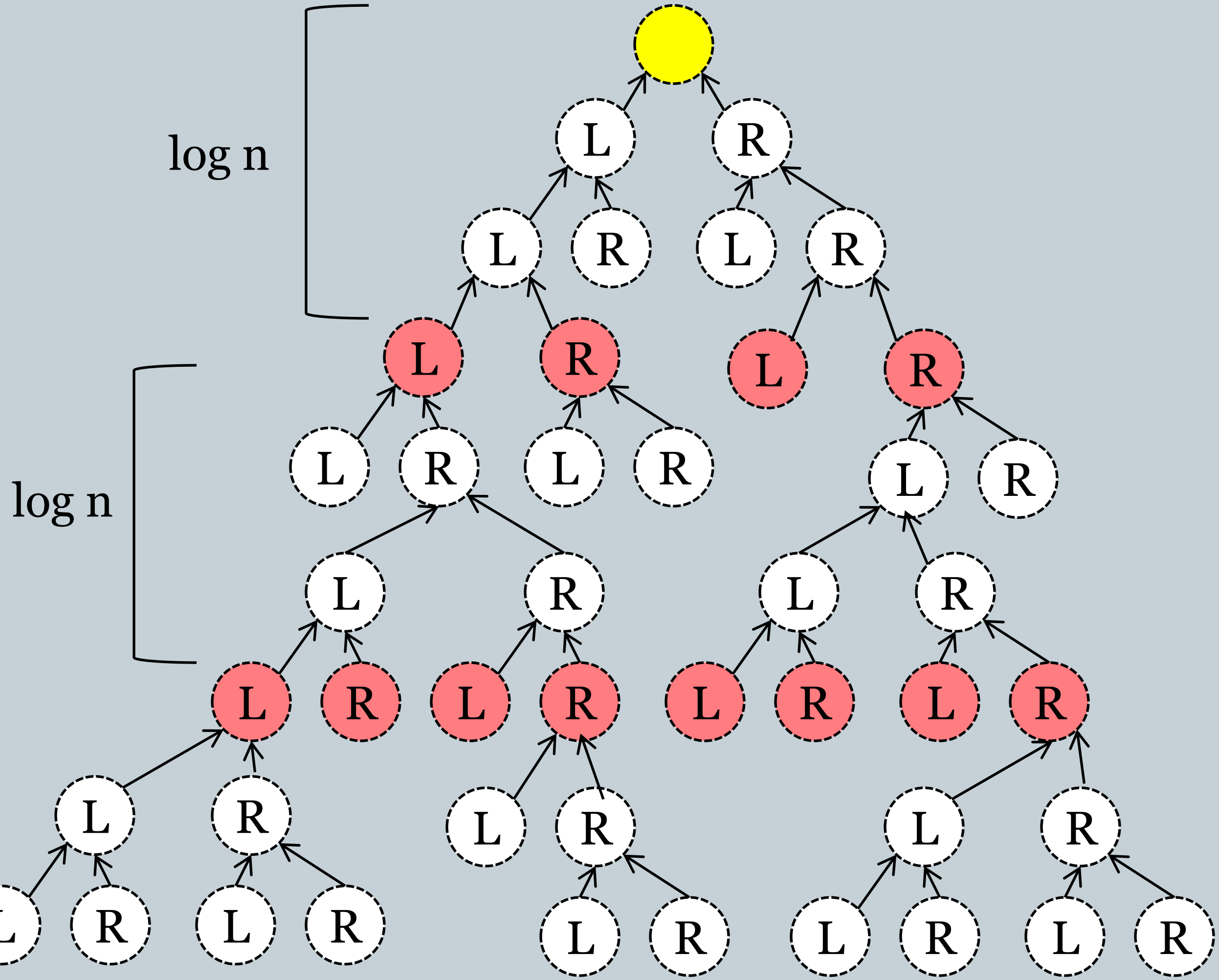


works, but takes too many queries

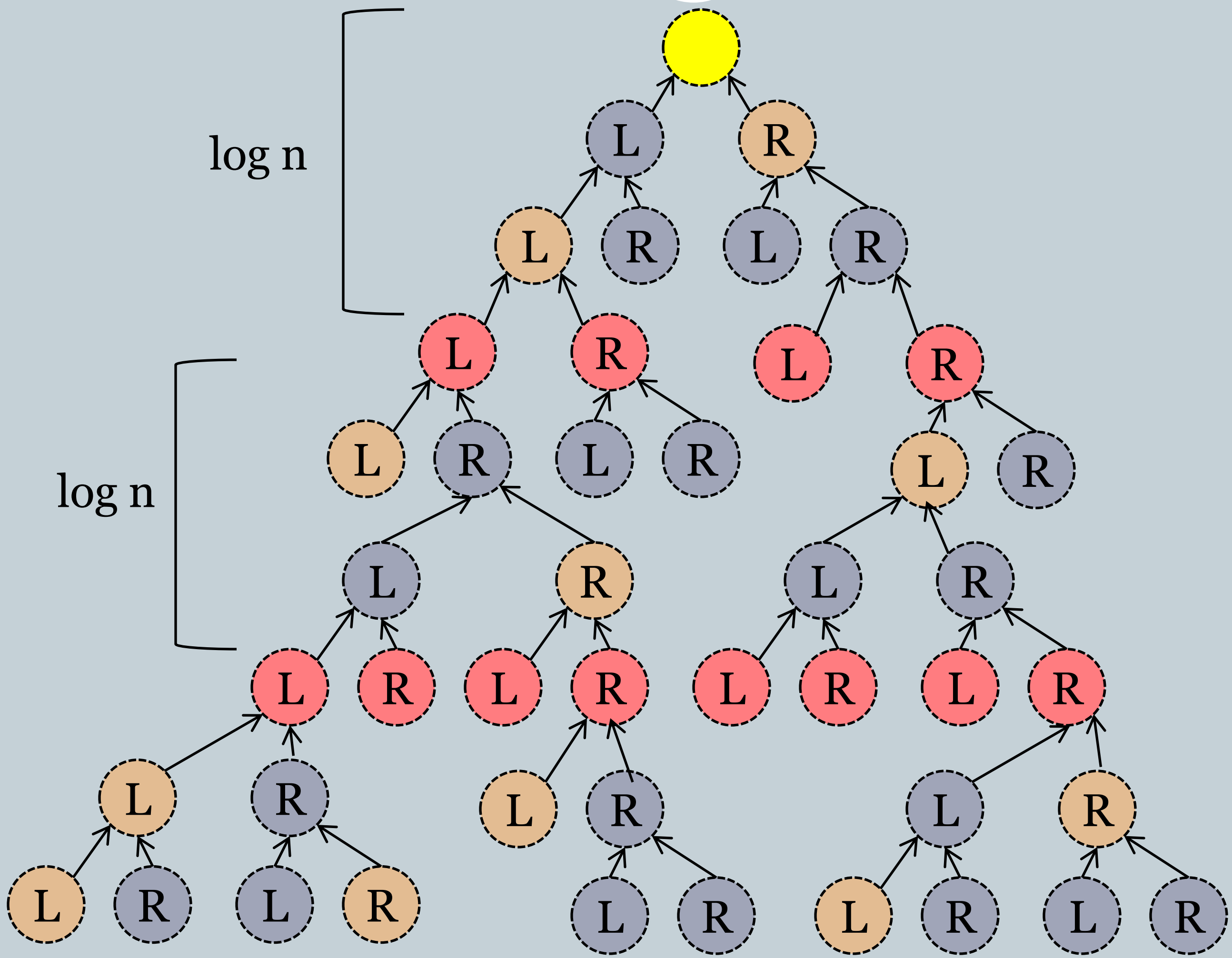
Result for c-concentrating Automata



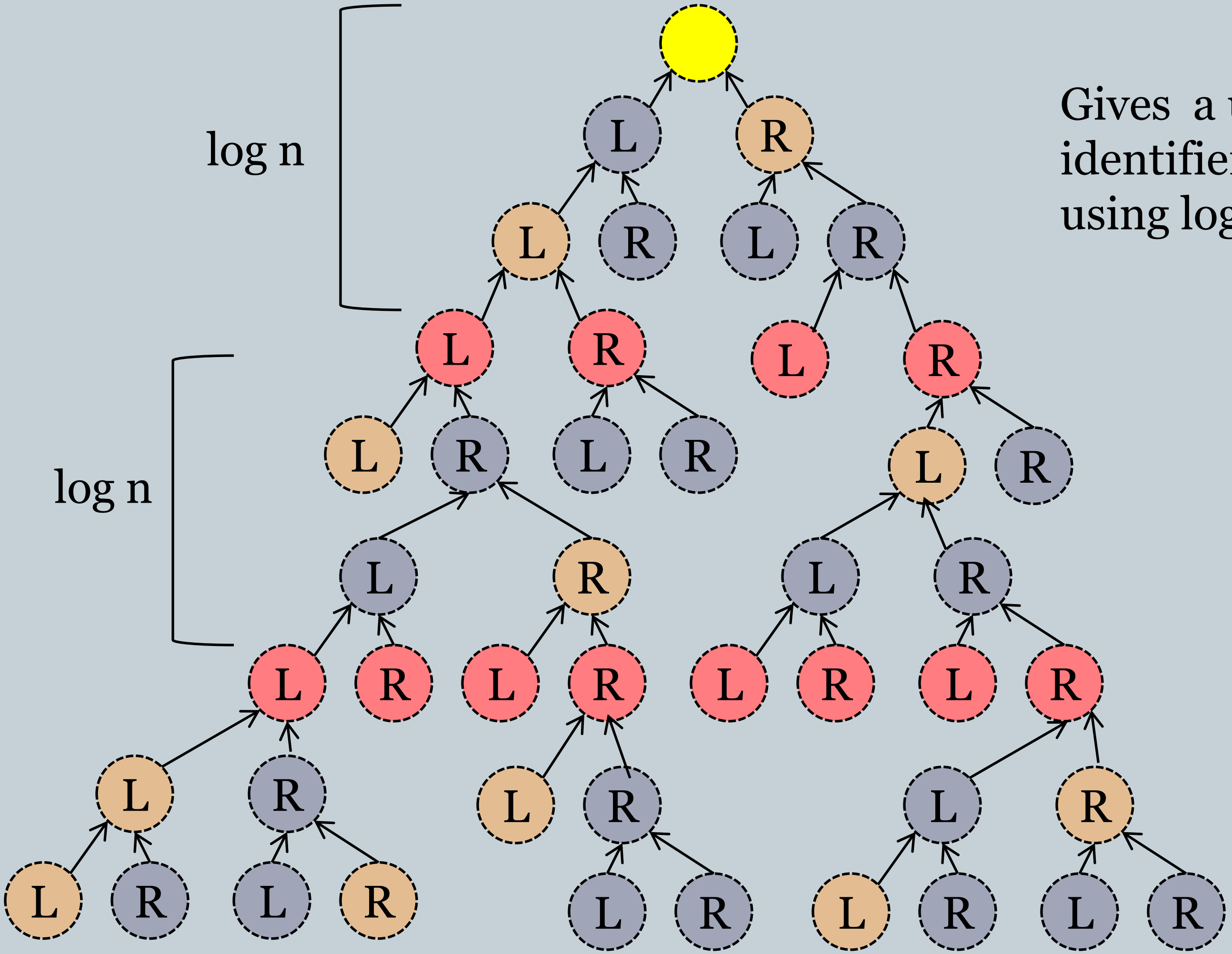
Result for c-concentrating Automata



Result for c-concentrating Automata



Result for c-concentrating Automata



Gives a unique identifier for each node using log n queries.

Result for c-concentrating Automata

- Remember, we have the result that if each state has a unique label, can do learning in $|X|\ n$ queries.

- We saw it takes $O(\log n)$ queries to get unique labels.

- Gives an $O(|X|\ n \log n)$ algorithm.

Results For Random Teachers

For a machine on n states, with input alphabet X, and output alphabet Y

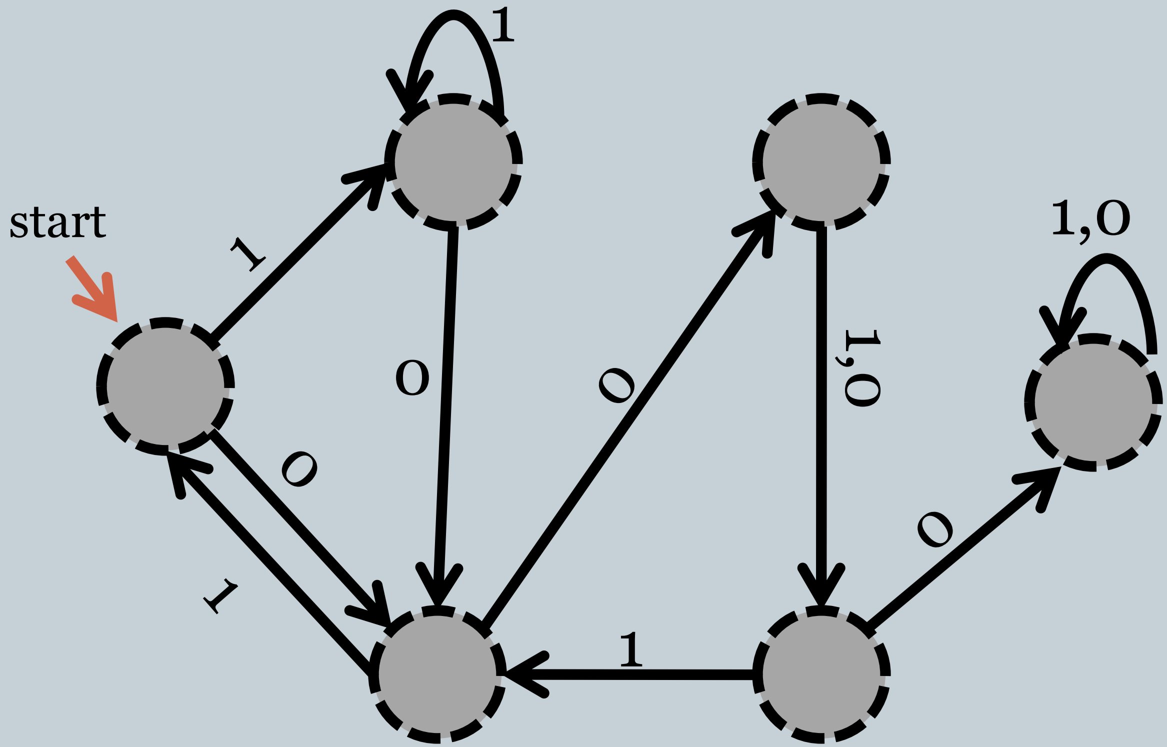
- A random teacher:
  - For any s, a teacher can use $|L| = |X|^s$ labels so that the learner uses $O(|X|n^{1+4/s}/\delta^{2/s})$ labels and succeeds w.p. $\geq 1-\delta$
    - so we can use $O(|X|^{4/\varepsilon})$ labels to need only $|X|n^{1+\varepsilon}$ queries
- A random automaton:
  - can be learned using O(n log n) label queries. Known from [Korshunov '67] but we give a different proof.
- Various results on unfolding and labeling:
  - one automaton can be transformed into an equivalent automaton before labels are placed. We call this "unfolding".

Results For Random Teachers

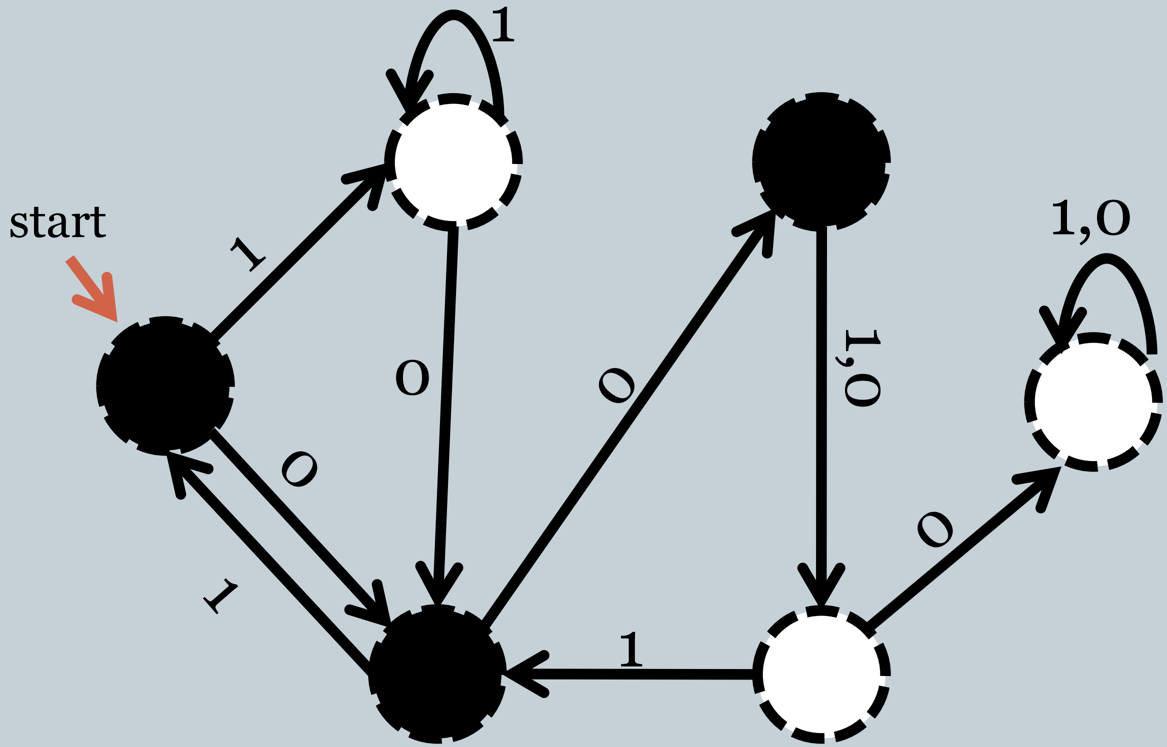
For a machine on n states, with input alphabet X, and output alphabet Y

- A random teacher:
  - For any s, a teacher can use $|L| = |X|^s$ labels so that the learner uses $O(|X|n^{1+4/s}/\delta^{2/s})$ labels and succeeds w.p. $\geq 1-\delta$
    - so we can use $O(|X|^{4/\varepsilon})$ labels to need only $|X|n^{1+\varepsilon}$ queries
- A random automaton:
  - can be learned using O(n log n) label queries. Known from [Korshunov '67] but we give a different proof.
- Various results on unfolding and labeling:
  - one automaton can be transformed into an equivalent automaton before labels are placed. We call this "unfolding".

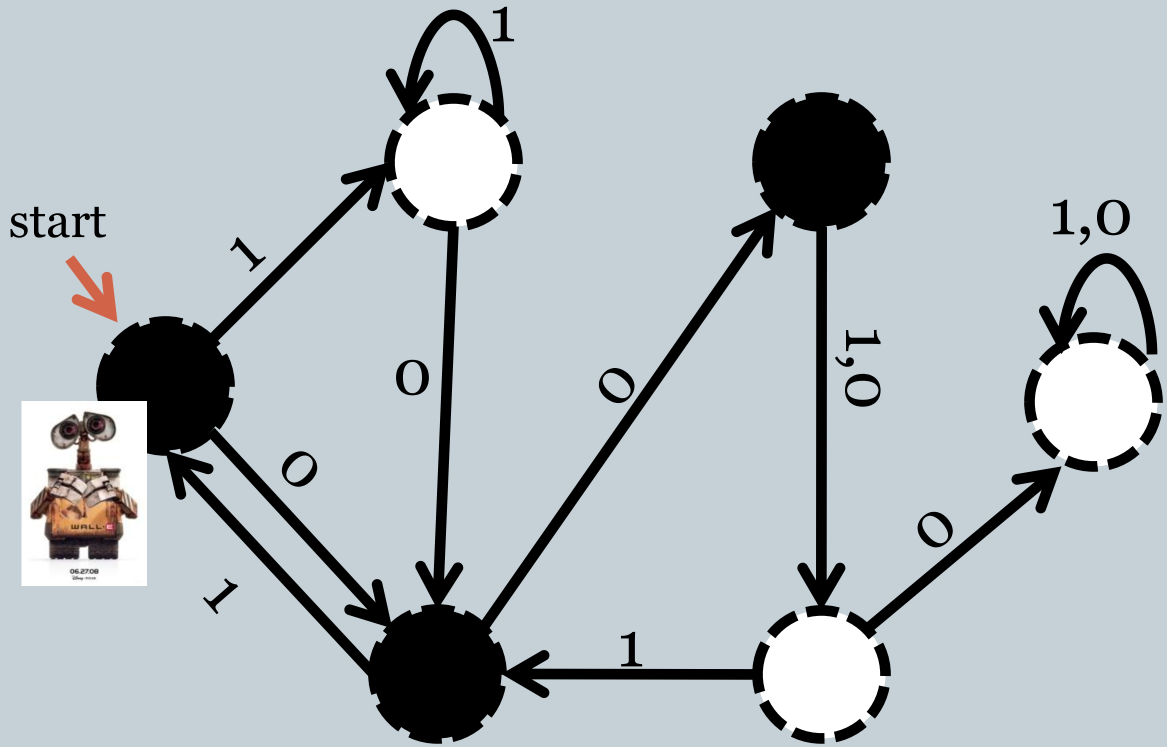
Using Results from [Freund et al. '97]



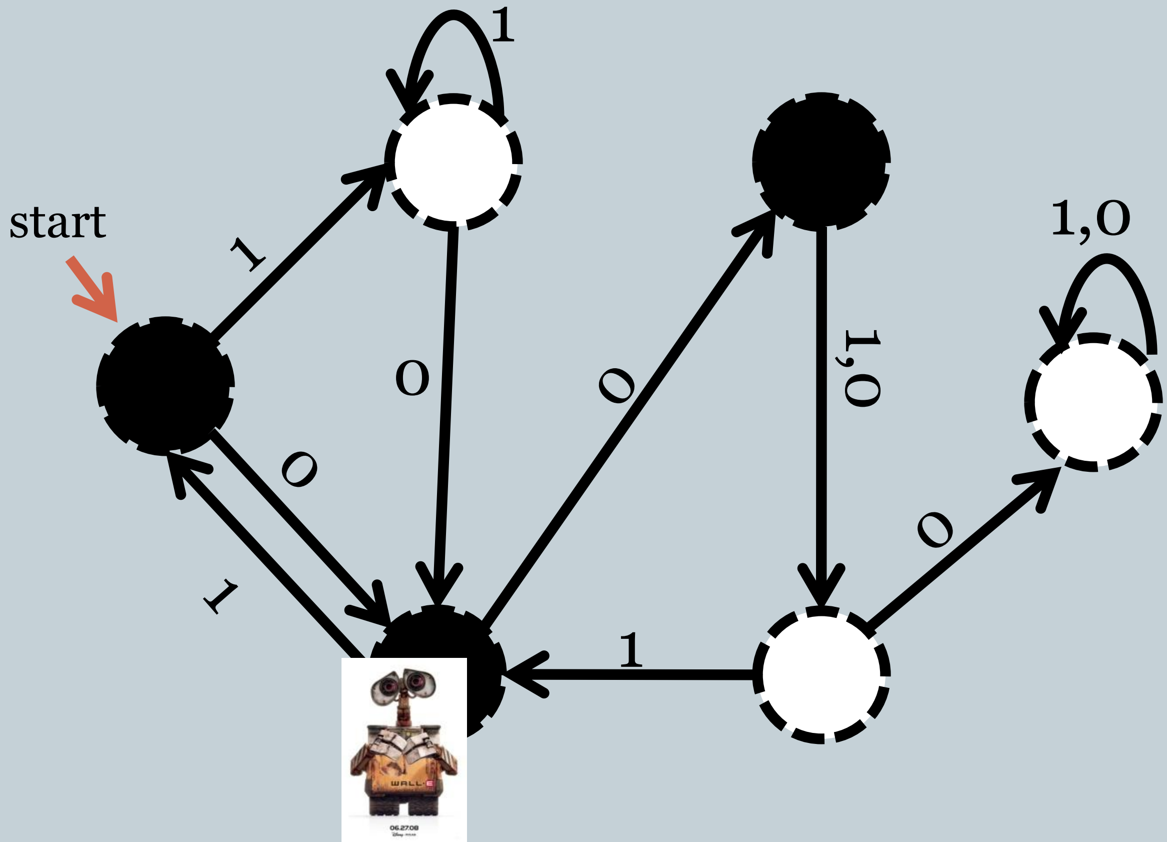
Using Results from [Freund et al. '97]



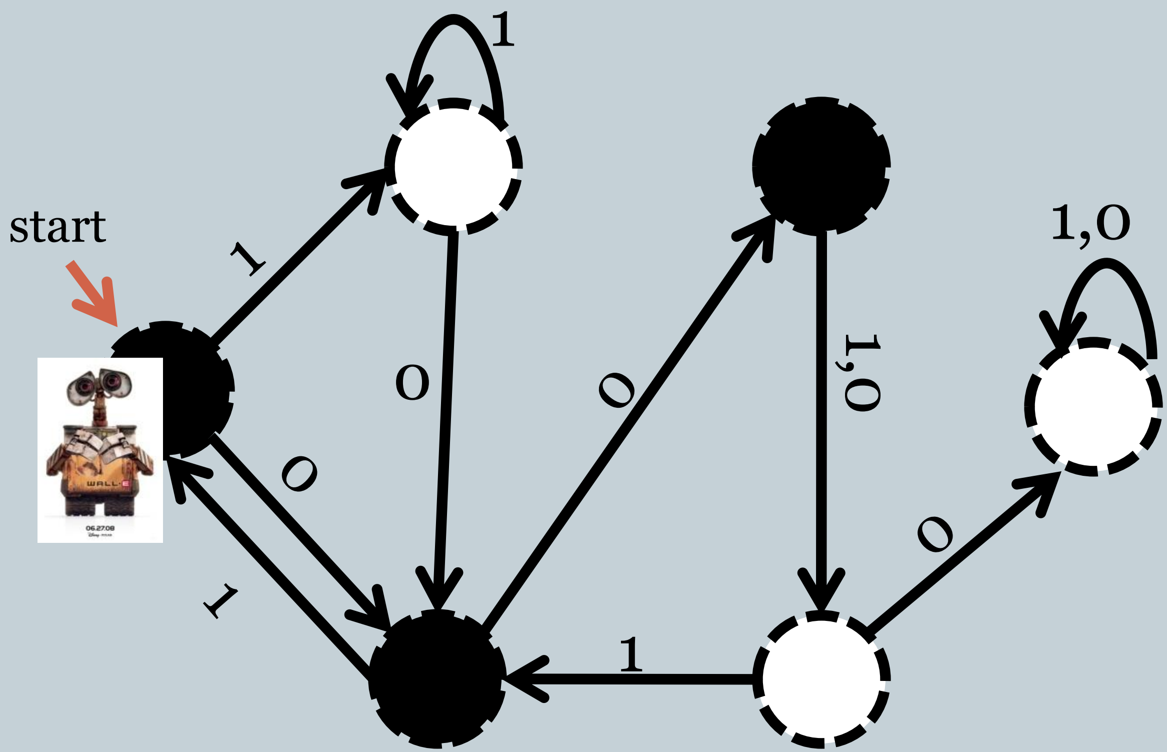
Using Results from [Freund et al. '97]



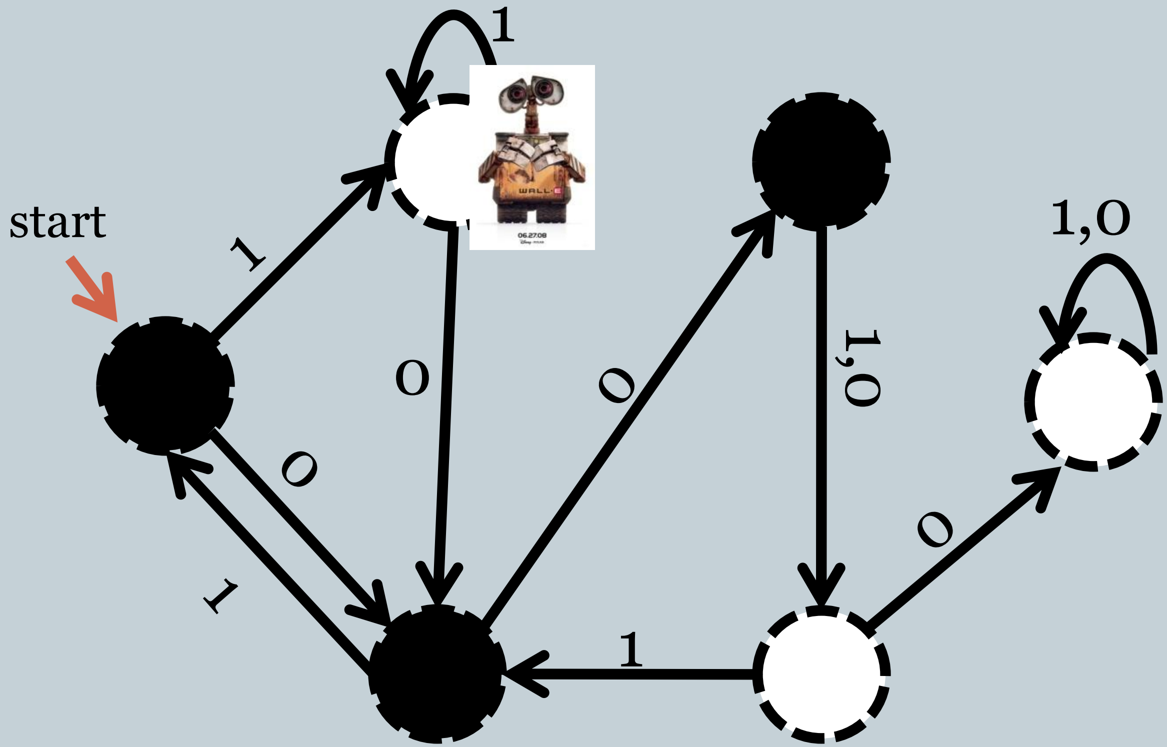
Using Results from [Freund et al. '97]



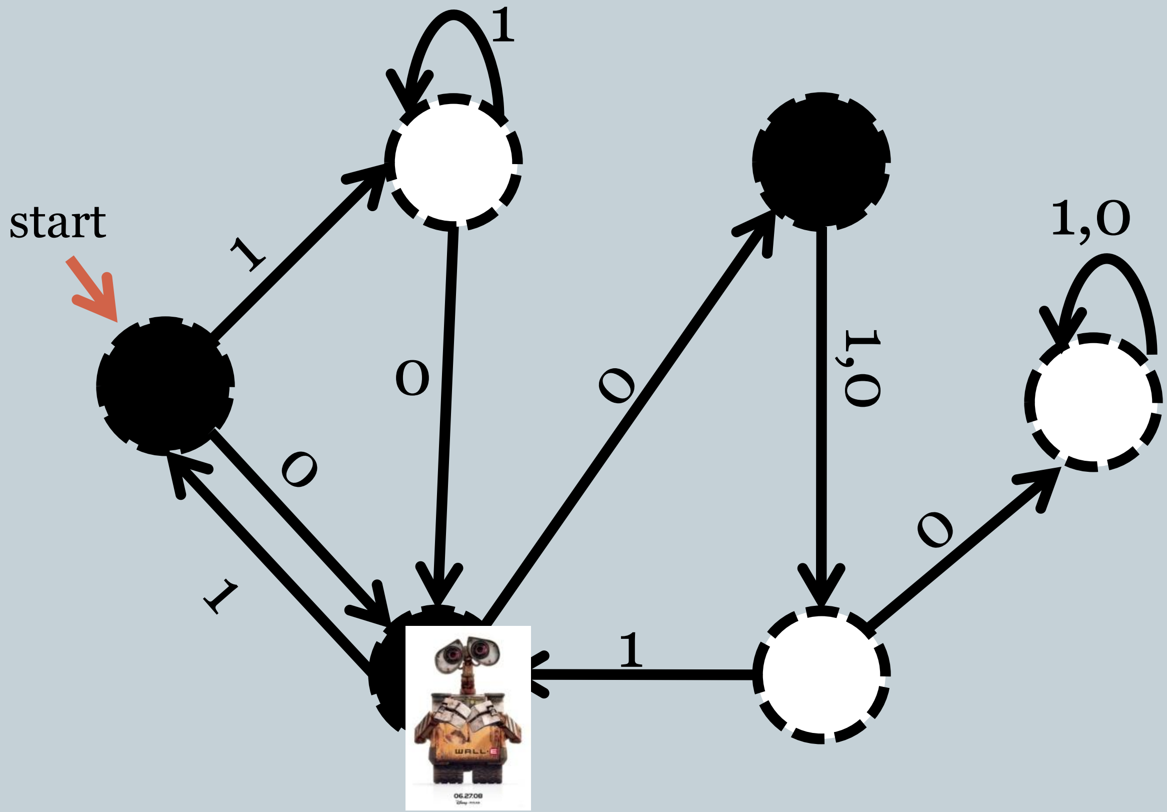
Using Results from [Freund et al. ’97]



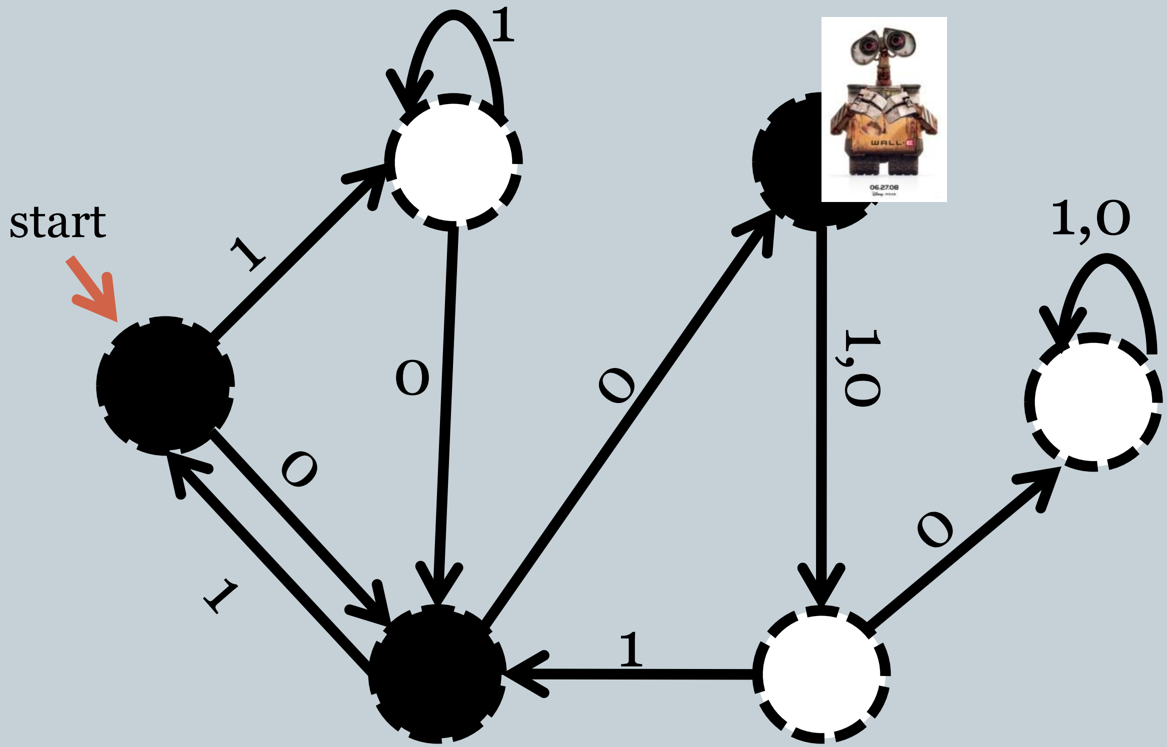
Using Results from [Freund et al. ’97]



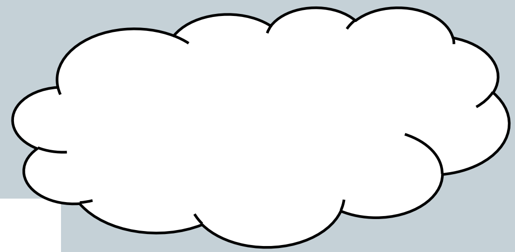
Using Results from [Freund et al. '97]

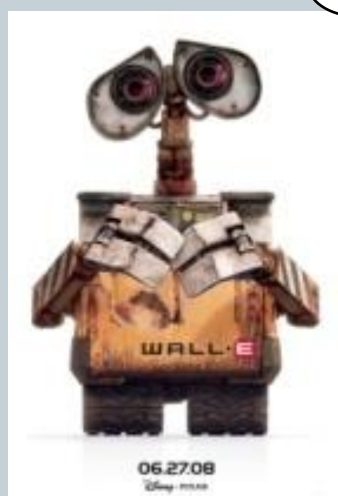


Using Results from [Freund et al. ’97]

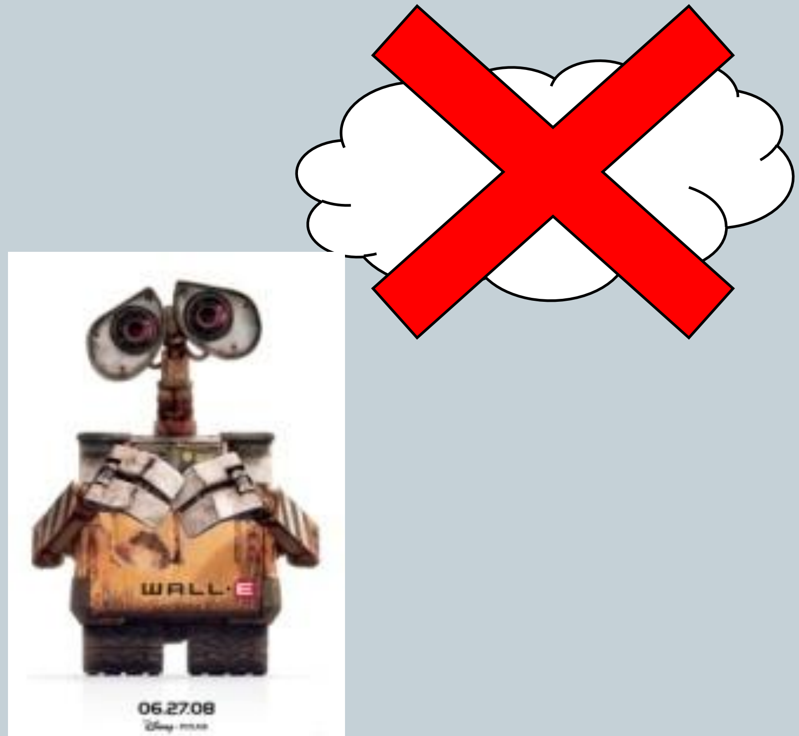


What the Robot Sees

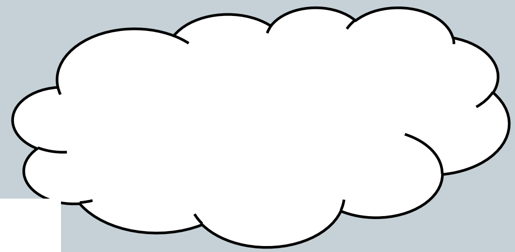




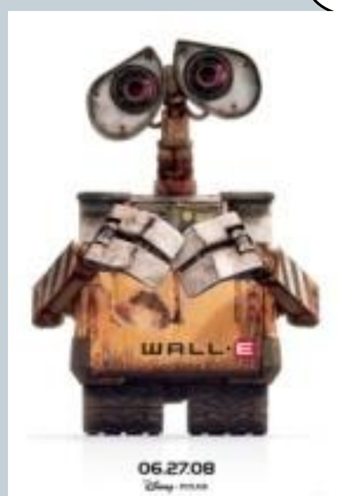
What the Robot Sees



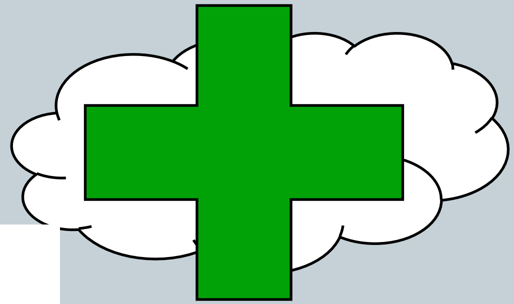
What the Robot Sees

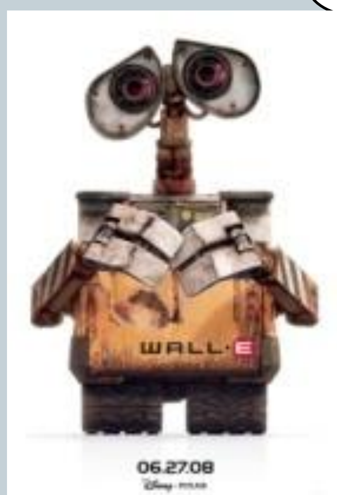


1



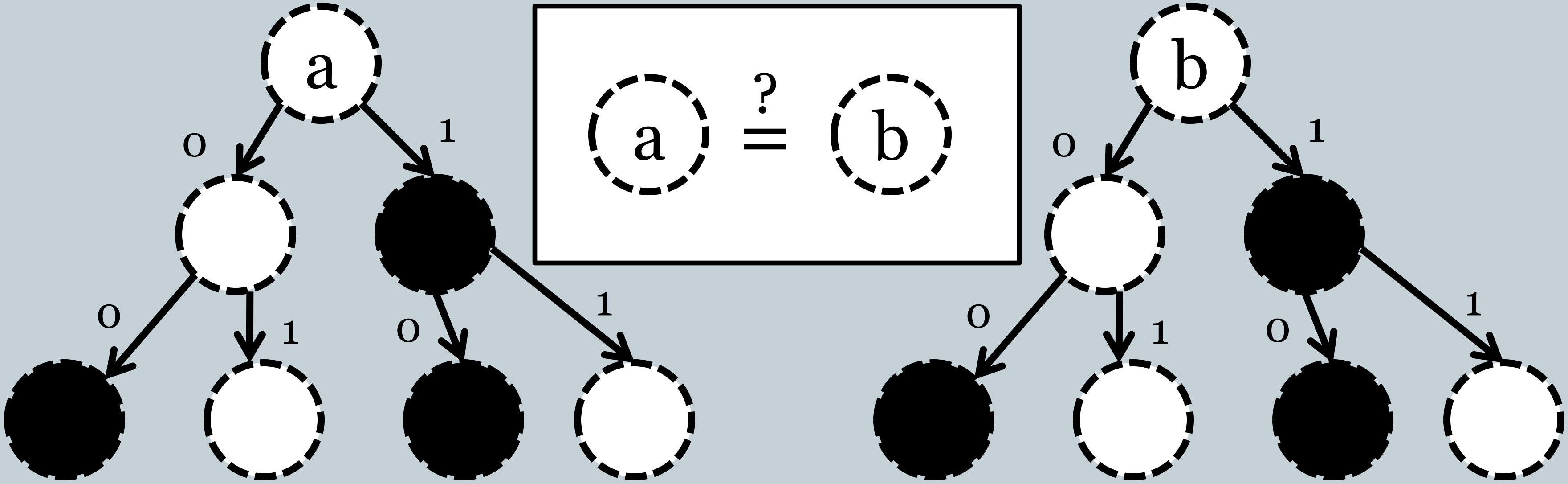
What the Robot Sees





Random Labeling of Automata

- [Freund et al. '97] show how to limit the number of prediction mistakes the robot makes to ~ $O(n^5)$.
- Their result makes use of signature trees.



Random Labeling of Automata

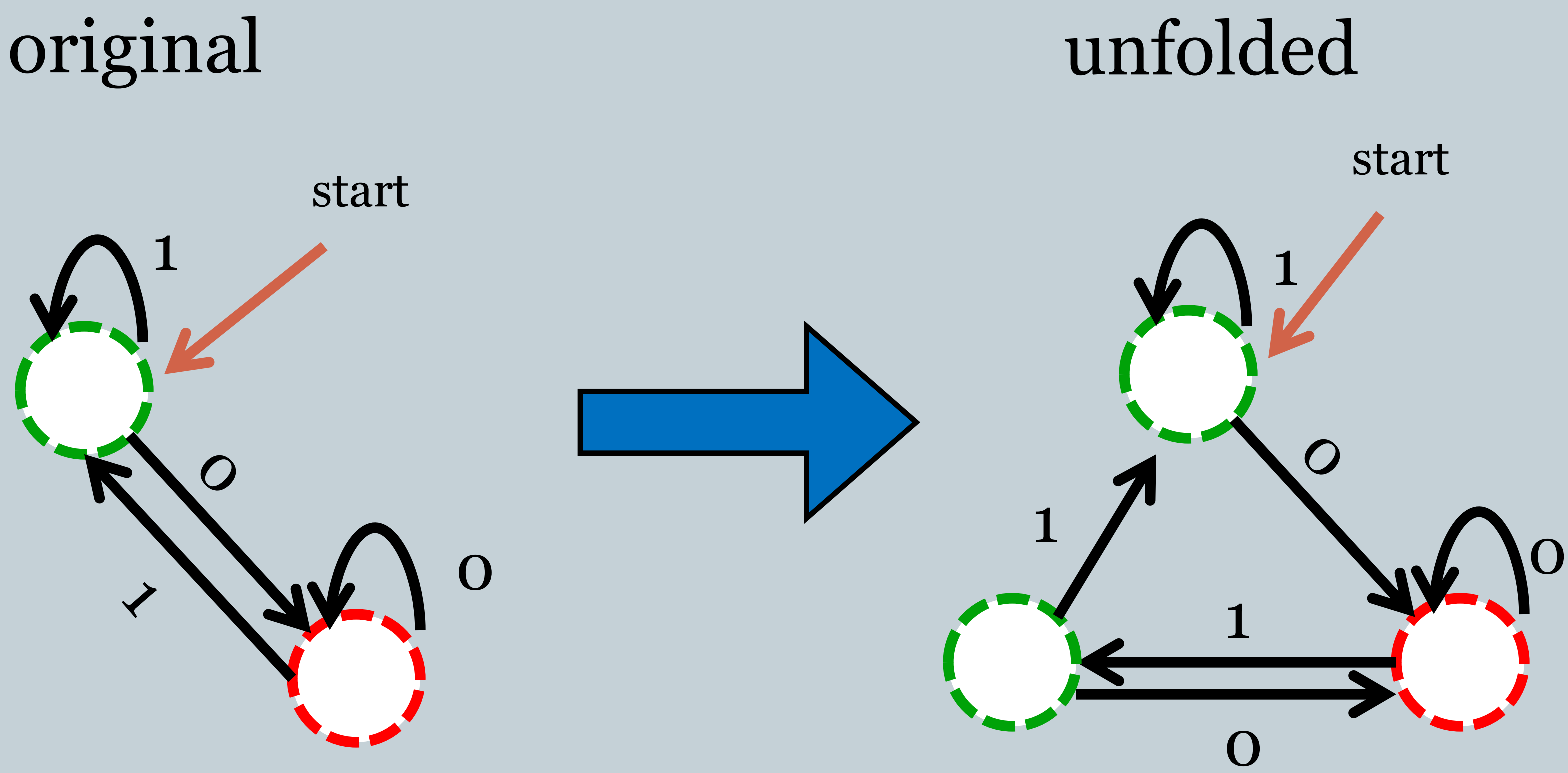
- [Freund et al. '97] show how to limit the number of prediction mistakes the robot makes to ~ $O(n^5)$.
- Their result makes use of signature trees.
- We extend their technique to the label query setting.
- We analyze this problem for an arbitrary number of labels.

Results For Random Teachers

For a machine on n states, with input alphabet X, and output alphabet Y

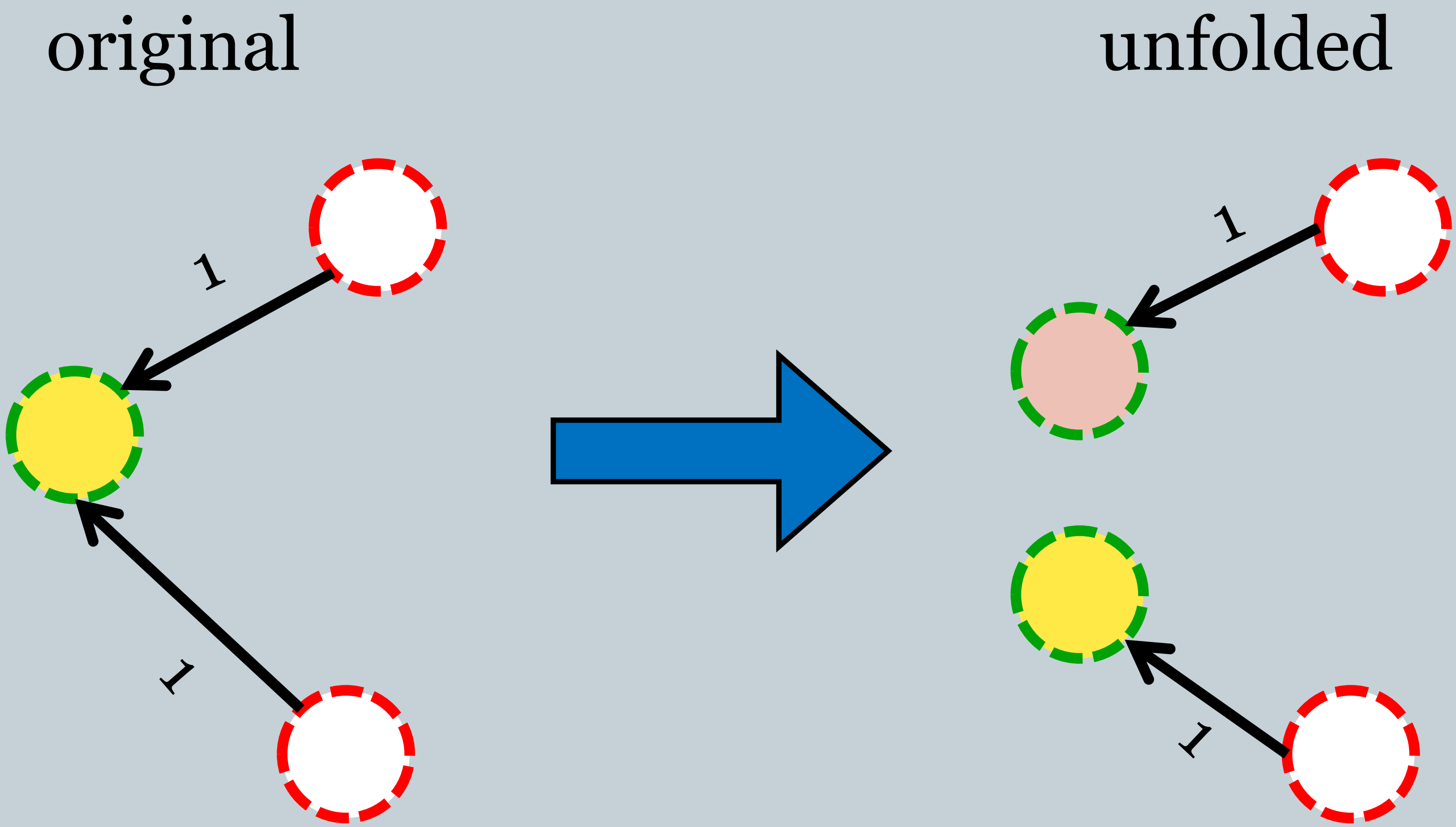
- A random teacher:
  - For any s, a teacher can use $|L| = |X|^s$ labels so that the learner uses $O(|X|n^{1+4/s}/\delta^{2/s})$ labels and succeeds w.p. $\geq 1-\delta$
    - so we can use $O(|X|^{4/\varepsilon})$ labels to need only $|X|n^{1+\varepsilon}$ queries
- A random automaton:
  - can be learned using O(n log n) label queries. Known from [Korshunov '67] but we give a different proof.
- Various results on unfolding and labeling:
  - one automaton can be transformed into an equivalent automaton before labels are placed. We call this "unfolding".

Unfolding an Automaton



model: teacher unfolds machine, then (randomly) labels it, and learner must use label queries to produce an output equivalent machine.

Why Unfolding Helps in Learning with Labels



Conclusions

- We analyzed learning with label queries: which are more powerful than membership queries.
- We have results on:
  - learning with helpful labels
  - learning with random labels
  - unfolding and random labels (unfolding w/ helpful labels is uninteresting because of coding tricks)
  - learning DFAs with random structure
- Most interesting open problem: can we learn DFA with a constant number of labels, helpfully placed, using $O(n \log n)$ queries?, to match the lower bound.
  - Our result is $O(n^2)$, using the live complete sample