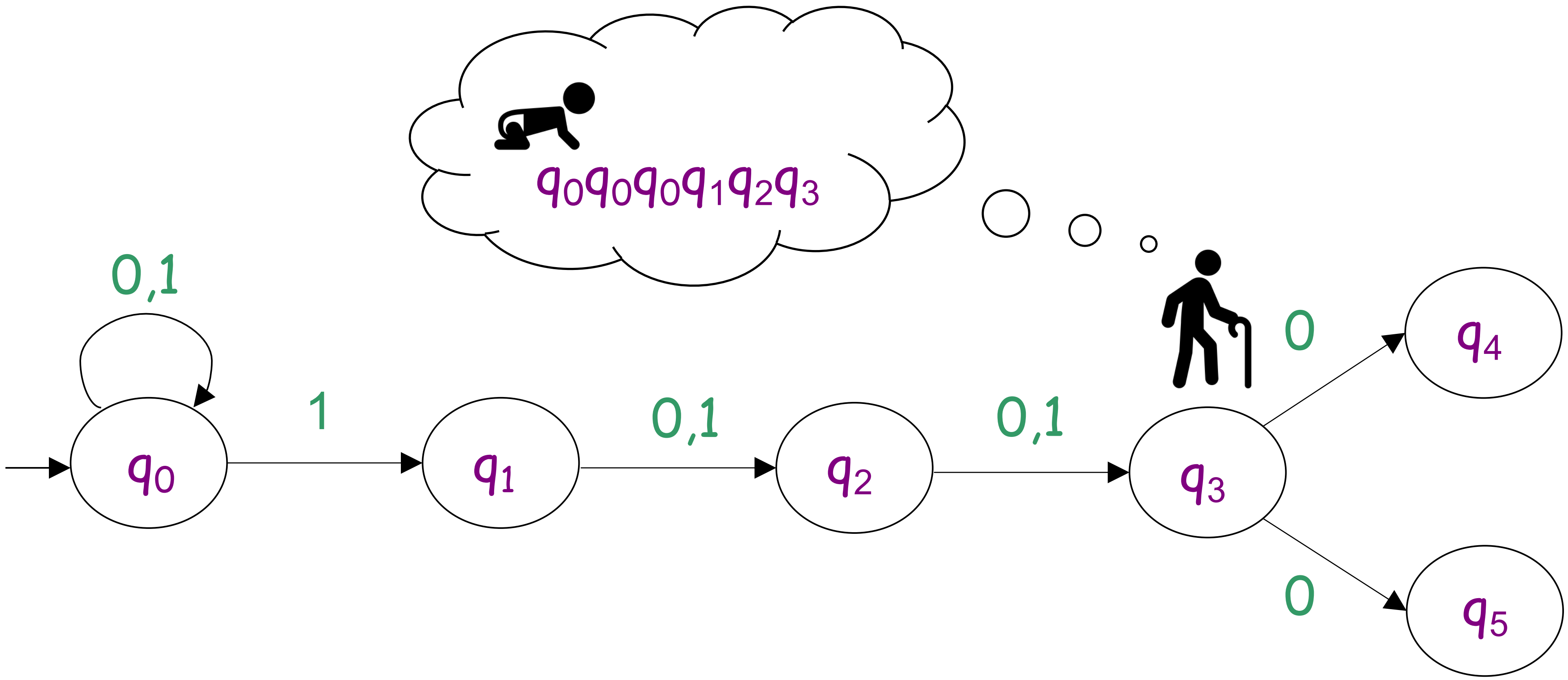
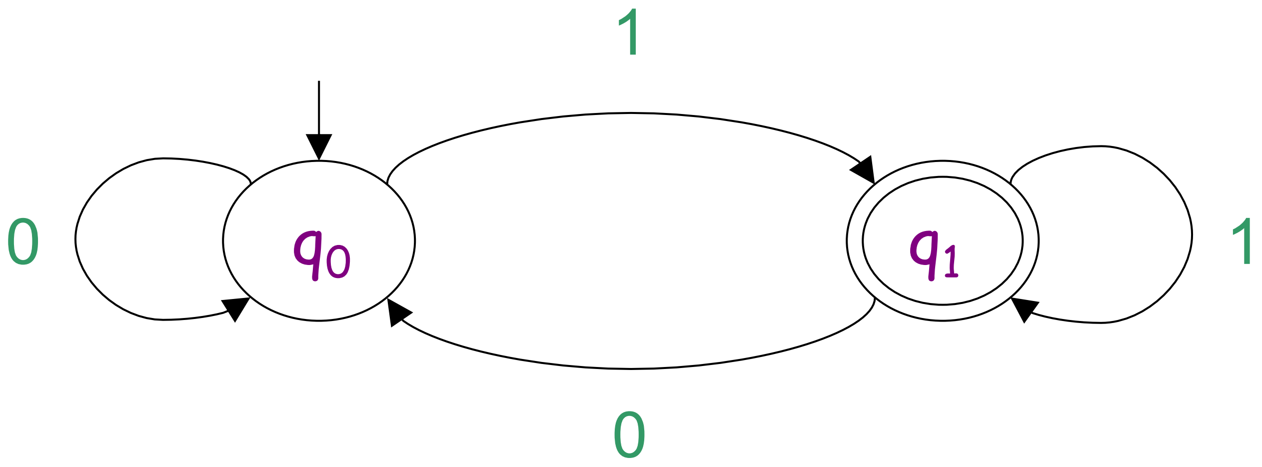
Using the past for resolving the future



Orna Kupferman

The Hebrew University

Automata



A=⟨Σ,Q,q_0,δ,α⟩ deterministic: δ: Q × Σ → Q

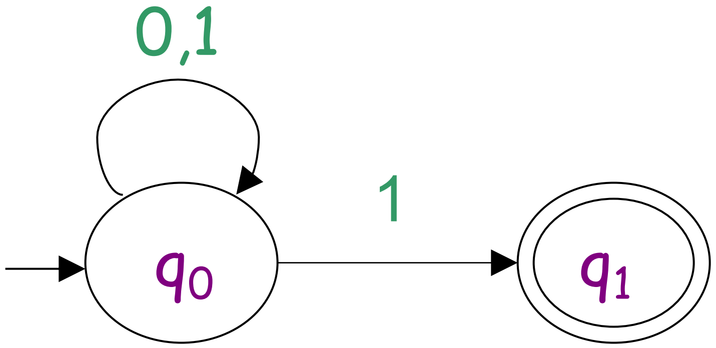
Finite words: a run is accepting if it ends in a state in α

L(A) = {w : w ends with 1}

Rabin Scott 1959: Finite automata and their decision problems

In Chapter II we consider possible generalizations of the notion of an automaton. A *nondeterministic automaton* has, at each stage of its operation, several choices of possible actions.

Nondeterministic Automata



$A = \langle \Sigma, Q, Q_0, \delta, \alpha \rangle$

non-deterministic: $\delta : Q \times \Sigma \rightarrow 2^Q$

L(A) = {w : w ends with 1}

Rabin Scott 1959: Finite automata and their decision problems

One might imagine at first sight that these new machines are more general than the ordinary ones, but this is not the case. We shall give a direct construction of an ordinary automaton, defining exactly the same set of tapes as a given nondeterministic machine.

Definition 11. *Let* $\mathfrak{A}=(S,M,S_0,F)$ *be a nondeterministic automaton.* $\mathfrak{D}(\mathfrak{A})$ *is the system* (T,N,t_0,G) *where* T *is the set of all subsets of* S, N *is a function on* $T\times\Sigma$ *such that* $N(t,\sigma)$ *is the union of the sets* $M(s,\sigma)$ *for* s *in* t, $t_0=S_0$, *and* G *is the set of all subsets of* S *containing at least one member of* F.

Clearly $\mathfrak{D}(\mathfrak{A})$ is an ordinary automaton, but it is actually equivalent to $\mathfrak{A}$.

Rabin Scott 1959: Finite automata and their decision problems

$\mathcal{U} \rightarrow \mathcal{D}(\mathcal{U})$

$\mathcal{U}$ is in one of the states $s_1, s_2, \ldots, s_k$ after reading w

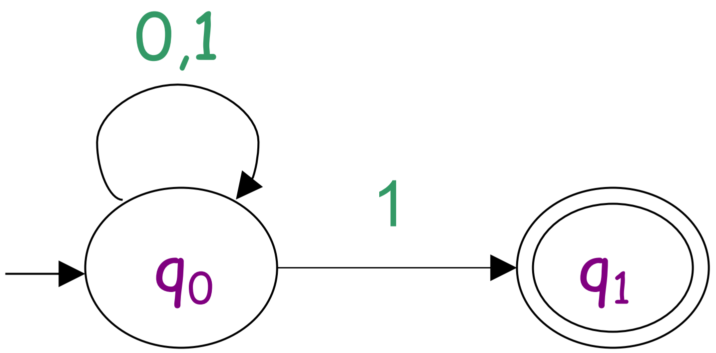
The subset construction

$\mathcal{D}(\mathcal{U})$ is in the state $\{s_1, s_2, \ldots, s_k\}$ after reading w

Definition 11. *Let $\mathfrak{A} = (S, M, S_0, F)$ be a nondeterministic automaton. $\mathfrak{D}(\mathfrak{A})$ is the system (T, N, t_0, G) where T is the set of all subsets of S, N is a function on $T \times \Sigma$ such that $N(t, \sigma)$ is the union of the sets $M(s, \sigma)$ for s in t, $t_0 = S_0$, and G is the set of all subsets of S containing at least one member of F.*

Clearly $\mathfrak{D}(\mathfrak{A})$ is an ordinary automaton, but it is actually equivalent to $\mathfrak{A}$.

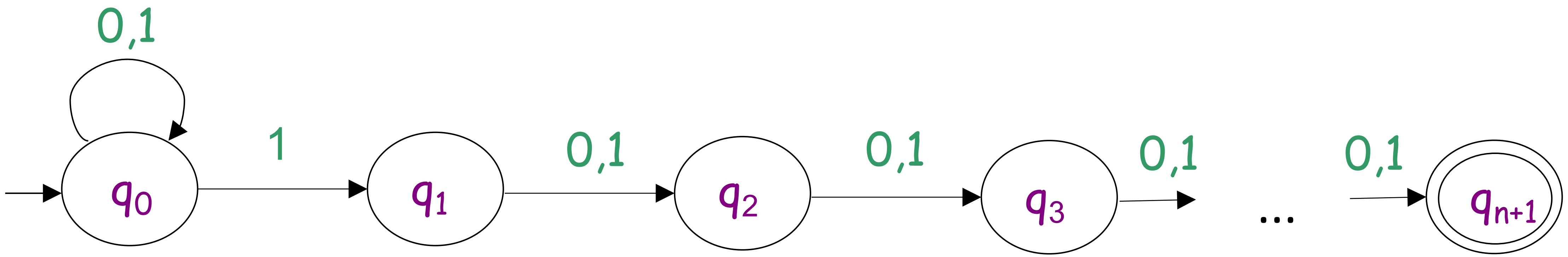
Nondeterministic Automata



$L(A_n) = \{w : w \text{ ends with } 1\cdot(0+1)^n\}$

$L(A) = \{w : w \text{ ends with } 1\}$

Nondeterministic Automata



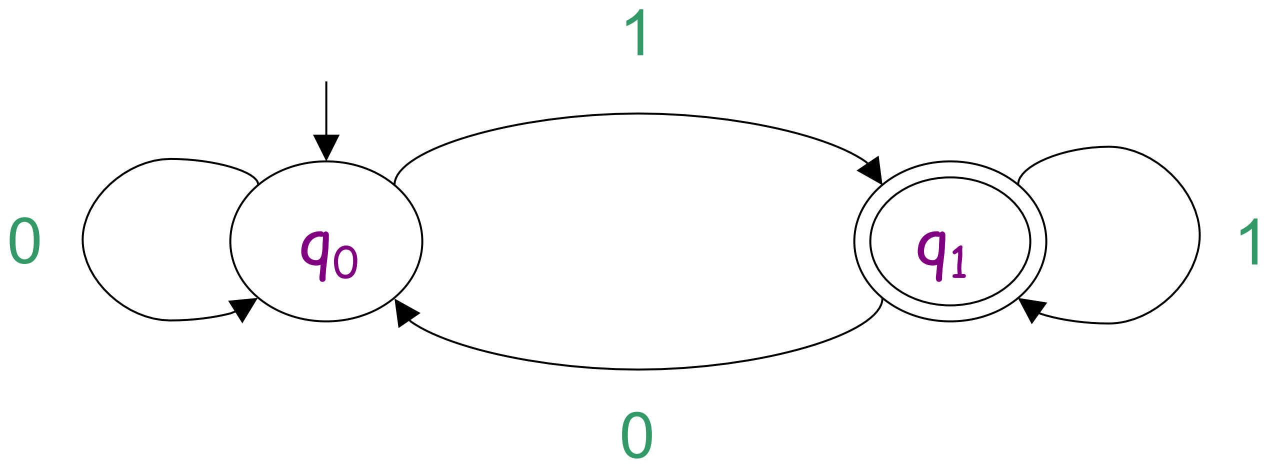
$L(A_n) = \{w : w \text{ ends with } 1\cdot(0+1)^n\}$

An equivalent deterministic automaton: 2^n states.

Finite words... easy!

Büchi 1962: Automata on infinite words

A run is accepting if it visits states in α infinitely often

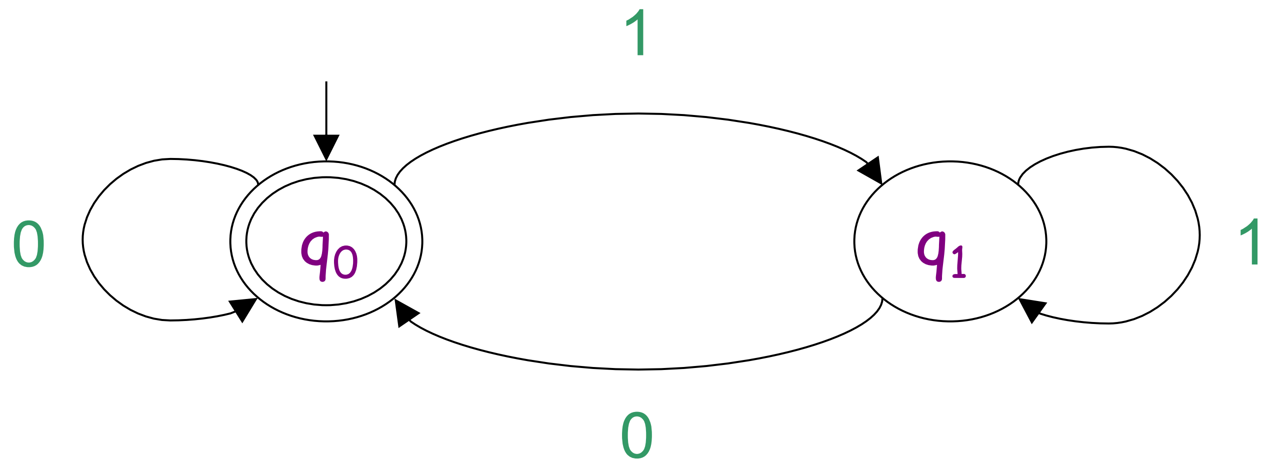


L(A) = {w : w has infinitely many 1s}

Complementation...

Büchi 1962: Automata on infinite words

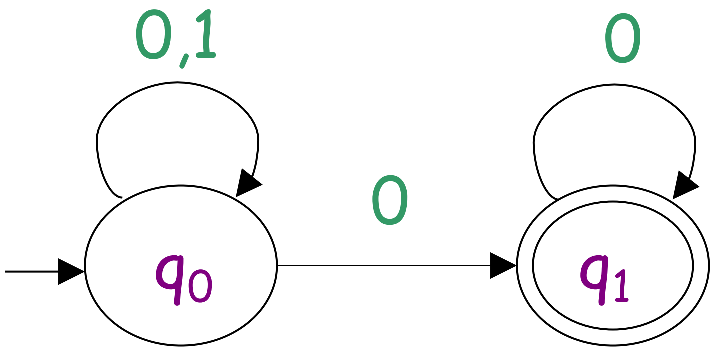
A run is accepting if it visits states in α infinitely often



L(A) = {w : w has infinitely many 0s}

Büchi 1962: Automata on infinite words

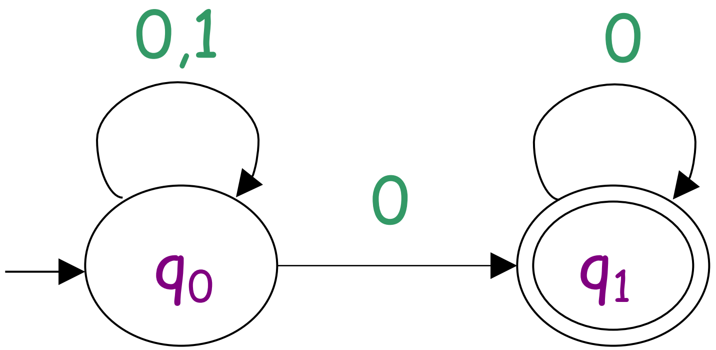
A run is accepting if it visits states in α infinitely often



L(A) = {w : w has finitely many 1s} = $(0+1)^* \cdot 0^\omega$

Büchi 1962: Automata on infinite words

A run is accepting if it visits states in α infinitely often



L(A) = {w : w has finitely many 1s}

Landweber 1969: no deterministic Büchi automaton!

So, NFW = DFW, yet NBW > DBW

- Three letter acronyms in {D,N} × {F,B,C,R,P} × {W,T}

{N,D}	branching mode (deterministic/nondeterministic)
{F,B,C,R,P}	acceptance condition (finite/Büchi/co-Büchi/Rabin/parity)
{W,T}	object to run on (words, trees)

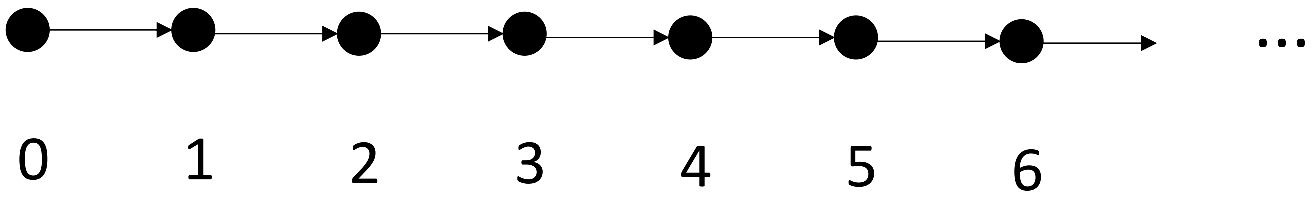
- NFW: Nondeterministic Finite Word automata
- DBW: Deterministic Büchi Word automata

Büchi 1962: Satisfiability of S1S ⟶ nonemptiness of NBWs.

- S1S: Monadic second order logic with one successor

$\forall x \exists y \; y > x$ $\quad x, y \in \mathbb{N}$

Every natural number has a bigger natural number.



Büchi 1962: S1S formula φ → NBW for the models of φ

Büchi 1962: Satisfiability of S1S $\longrightarrow$ nonemptiness of NBWs.

- S1S: Monadic second order logic with one successor

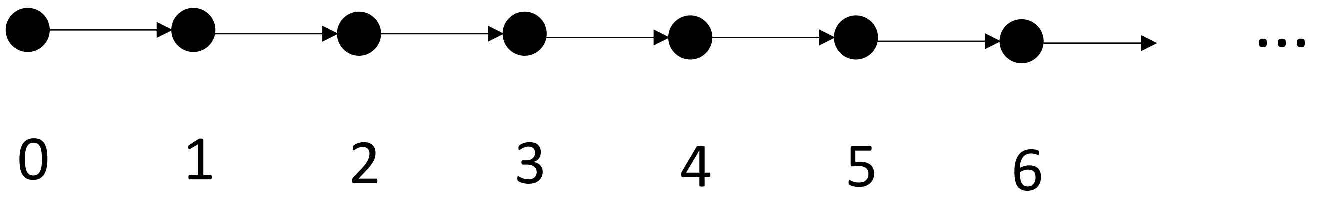
$\forall x \exists y \; y>x \; \wedge P_a(y)$

$x,y \in \mathbb{N}$

Words in $(a+b)^\omega$

$P_a, P_b \subseteq \mathbb{N}$

Infinitely many a's

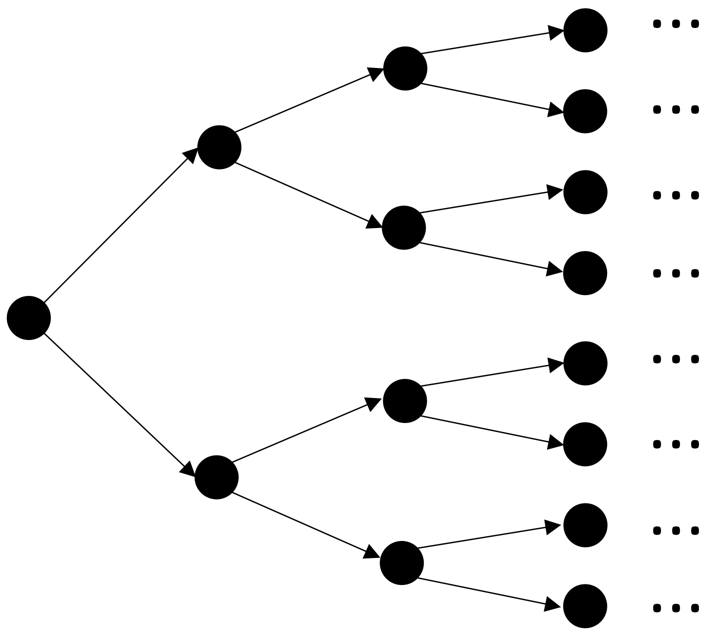


Büchi 1962: S1S formula φ $\rightarrow$ NBW for the models of φ

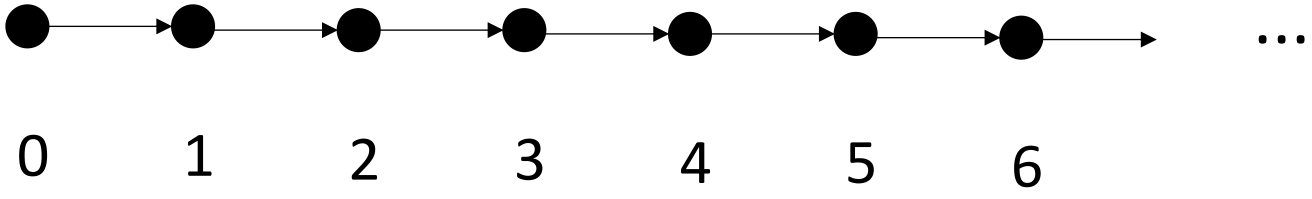
Can we do the same with SnS?

Can we do the same with SnS?

- SnS: Monadic second order logic with n successors

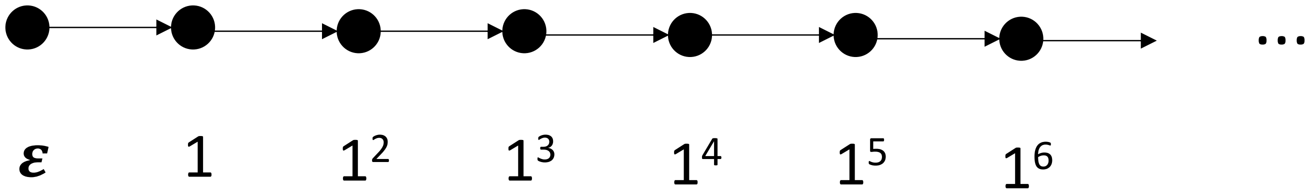


S2S: two successors

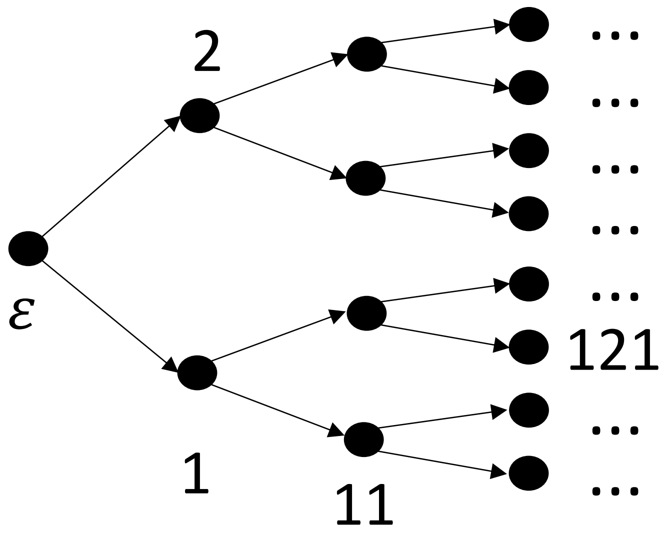


S1S: one successor

$x \in \mathbb{N}$



$x \in 1^*$



S2S: two successors

$x \in \{1,2\}^*$

- SnS: Monadic second order logic with n successors

Every non-root node in the tree {1,2}* has a single parent

$\forall x\ (x \neq \varepsilon) \rightarrow \exists y\ (x=y.1 \vee x=y.2) \wedge \forall z.\ (z \neq y) \rightarrow (x \neq z.1 \wedge x \neq z.2)$

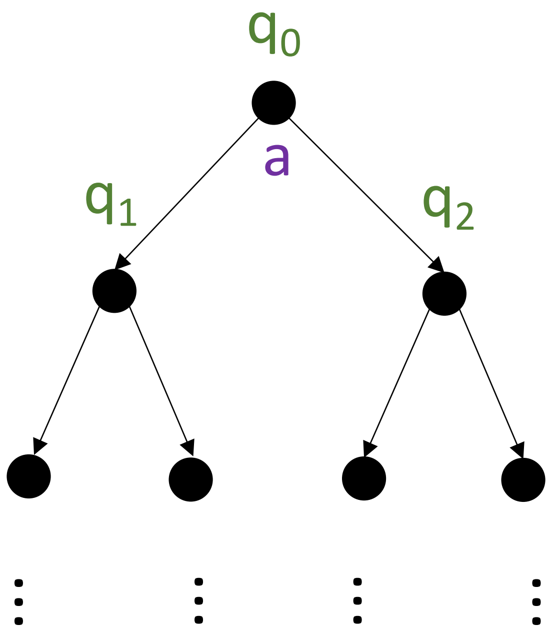
$x,y,z \in \{1,2\}^*$

- The community 1962+: let's reduce SnS ⟶ NBT

- NBT: Nondeterministic Büchi Tree automata

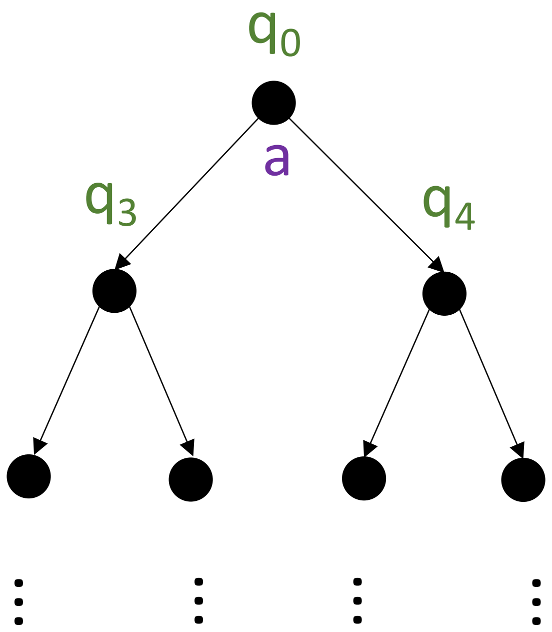
Tree automata:

$\delta(q_0,a)=\{\langle q_1,q_2\rangle,\langle q_3,q_4\rangle\}$



Tree automata:

$\delta(q_0,a)=\{\langle q_1,q_2\rangle,\langle q_3,q_4\rangle\}$



a run is accepting if it visits states in α infinitely often in all paths

Rabin 1969: Satisfiability of SnS $\longrightarrow$ nonemptiness of NRTs.

- NRT: Nondeterministic Rabin Tree automata
- A stronger acceptance condition

Rabin 1971:

- NRT > NBT
- No reduction SnS $\longrightarrow$ NBT
- And I knew it since 1969...

- So, Rabin 1971: NRT > NBT

- Witness: in all paths of the tree, only finitely many 1's.

- Proof: very complicated.

- Note: this is L^{Δ}, for L={w: w has only finitely many 1's}

$L \subseteq \Sigma^{\omega}$ L^{Δ} : Σ-labeled trees all whose paths are in L

- Kupferman Safra Vardi 1996:

A reminder:
$L = \{w : w \text{ has finitely many 1s}\}$

Landweber 69: no DBW for L
Rabin 71: no NBT for L^{Δ}

KSV96: Rabin's choice of L^{Δ} was not coincidental

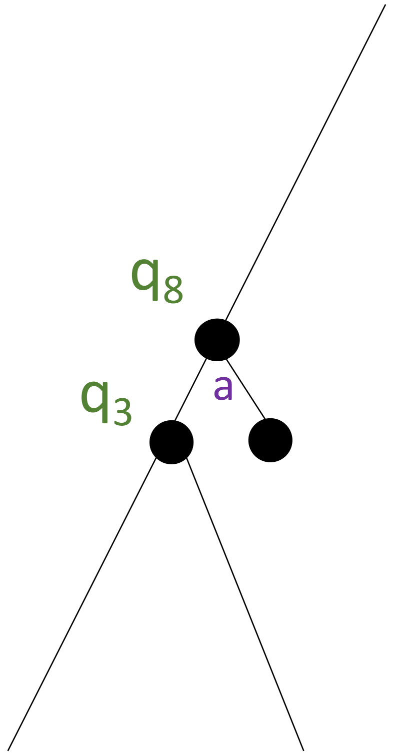
- KSV1996: $L \in NBW \setminus DBW$ iff $L^{\Delta} \in NRT \setminus NBT$

- Hard direction: L^{Δ} in NBT $\longrightarrow$ L in DBW

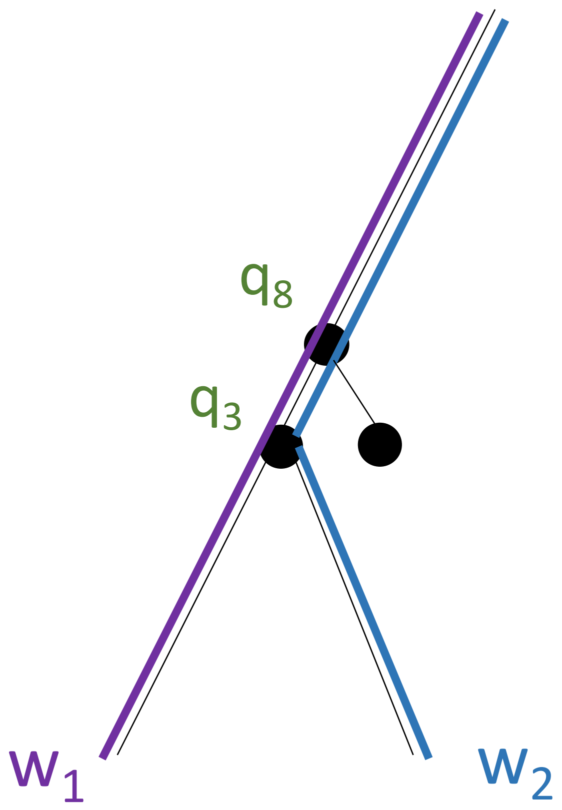
Intuition: a nondeterministic tree automaton for an L^{Δ} language cannot really guess

Intuition: a nondeterministic tree automaton for an L^Δ language cannot really guess

$$\delta(q_8,a)=\{\langle q_1,q_2\rangle,\langle q_3,q_4\rangle\}$$



Intuition: a nondeterministic tree automaton for an L^{Δ} language cannot really guess.



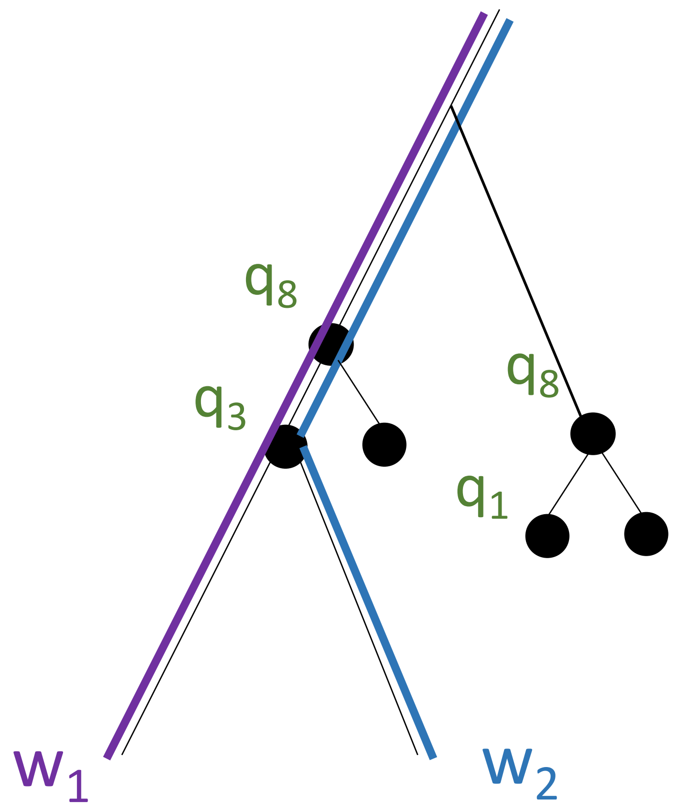
$\delta(q_8,a)=\{\langle q_1,q_2\rangle,\langle q_3,q_4\rangle\}$

- The guess to proceed with q_3 should work for both w_1 and w_2

- So a deterministic word automaton for L can have $\delta(q_8,a)=q_3$

- almost...

Intuition: a nondeterministic tree automaton for an L^{Δ} language cannot really guess.



$\delta(q_8,a)=\{\langle q_1,q_2\rangle,\langle q_3,q_4\rangle\}$

- The guess to proceed with q_3 should work for both w_1 and w_2

- So a deterministic word automaton for L can have $\delta(q_8,a)=q_3$

- almost... perhaps $\delta(q_8,a)=q_1$?

Kupferman Safra Vardi 1996:
NBT for L^{Δ} with state space Q $\longrightarrow$ DBW for L with state space 2^{Q}
(almost the subset construction).

Niwinski Walukiewicz 1998:
NXT for L^{Δ} $\longrightarrow$ DXW for L

- No extra expressive power
- What about succinctness?

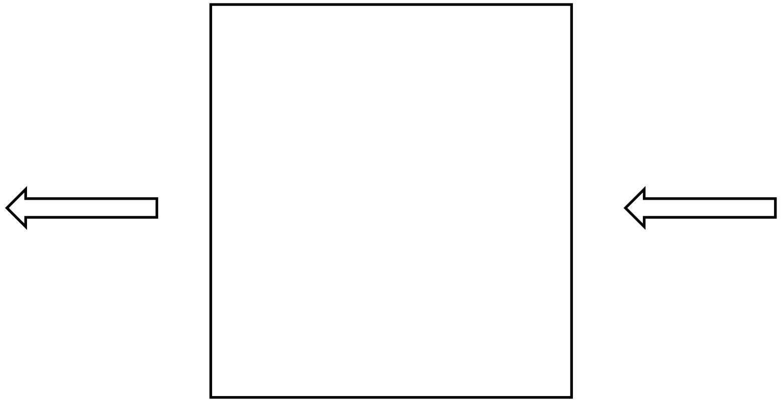
2000+: shift: tree automata ⟶ games

2000+: Shift: tree automata $\longrightarrow$ games

Main application: synthesis of reactive systems [Pnueli Rozner 1988]

A specification L $\subseteq (2^{I \cup O})^\omega$

$o_1, o_2, o_3, o_4, \ldots \in (2^O)^\omega$



$i_1, i_2, i_3, i_4, \ldots \in (2^I)^\omega$

2000+: Shift: tree automata $\longrightarrow$ games

Main application: synthesis of reactive systems [Pnueli Rozner 1988]

A specification L $\subseteq (2^{I \cup O})^{\omega}$

Story 1: label by letters in 2^O a tree with directions in 2^I: branches correspond to inputs, labels to outputs $\longrightarrow$ nonemptiness of a tree automaton for L^{Δ}

Story 2: find a winning strategy in a game played on a deterministic automaton for L: environment chooses actions in 2^I, system responds with actions in 2^I $\longrightarrow$ game solving

Henzinger Piterman 2006: History Deterministic (HD) automata

Intuitively: good for games

Formally: $A=\langle\Sigma,Q,Q_0,\delta,\alpha\rangle$ is HD if there is $f\colon \Sigma^* \to Q$ such that

- $f(\varepsilon)\in Q_0$
- for all $w\in\Sigma^*$ and $\sigma\in\Sigma$, we have that $f(w\cdot\sigma)\in\delta(f(w),\sigma)$
- for all $w\in\Sigma^\omega$, if $w\in L(A)$, then f(w) is accepting

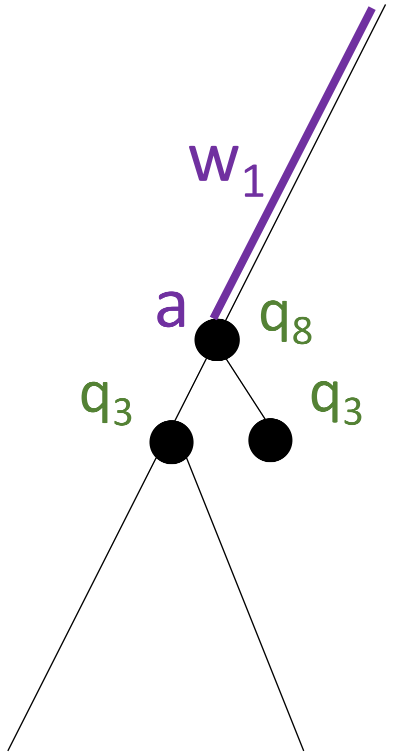
$f(w)=f(\varepsilon).f(\sigma_1).f(\sigma_1\sigma_2).\ f(\sigma_1\sigma_2\sigma_3)....$

Resolves nondeterminism in a way that only depends on the past

Resolves nondeterminism in a way that only depends on the past.

$\delta(q_8,a)=\{q_1,q_3,q_4\}$

- $f(w_1 \cdot a)=q_3$

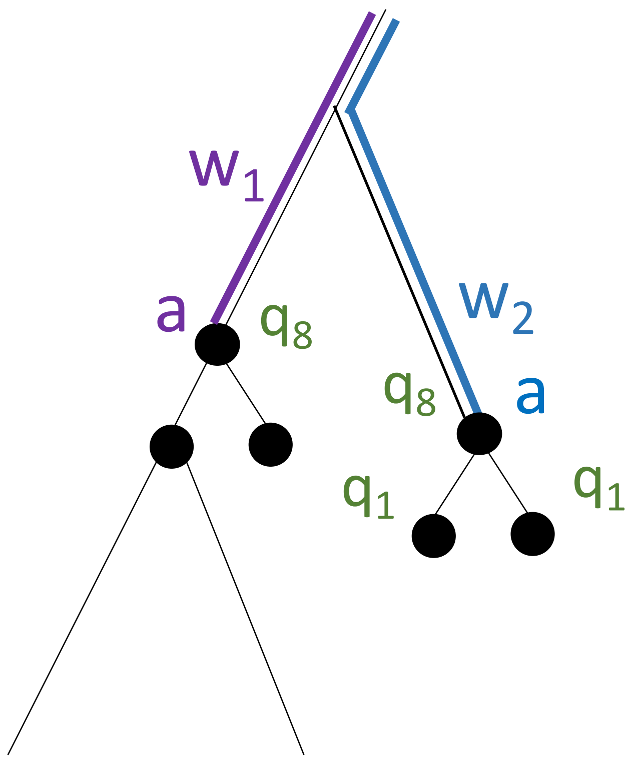


Σ-tree

Resolves nondeterminism in a way that only depends on the past.

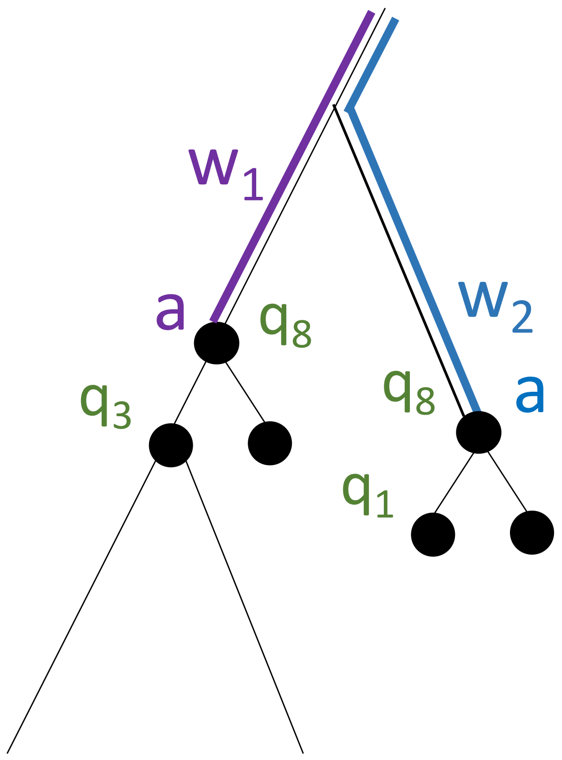
$\delta(q_8,a)=\{q_1,q_3,q_4\}$

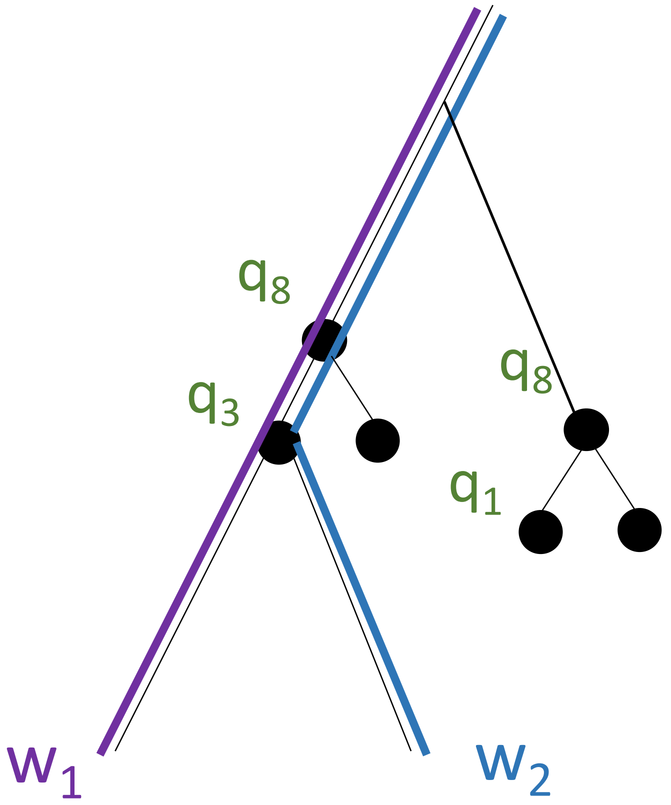
- $f(w_1 \cdot a)=q_3$
- $f(w_2 \cdot a)=q_1$



Σ-tree

Boker Kuperberg Kupferman Skrzypczak 2013: HD-NXW for L = NXT for L^{Δ}





GFG: Σ-tree, accepting runs in branches in L

tree in L^{Δ}: accepting runs in all branches

Kupferman Safra Vardi 1996:

NBT for L^{Δ} with state space Q $\longrightarrow$ DBW for L with state space 2^{Q} (almost the subset construction).

Kuperberg, Skrzypczak 2016:

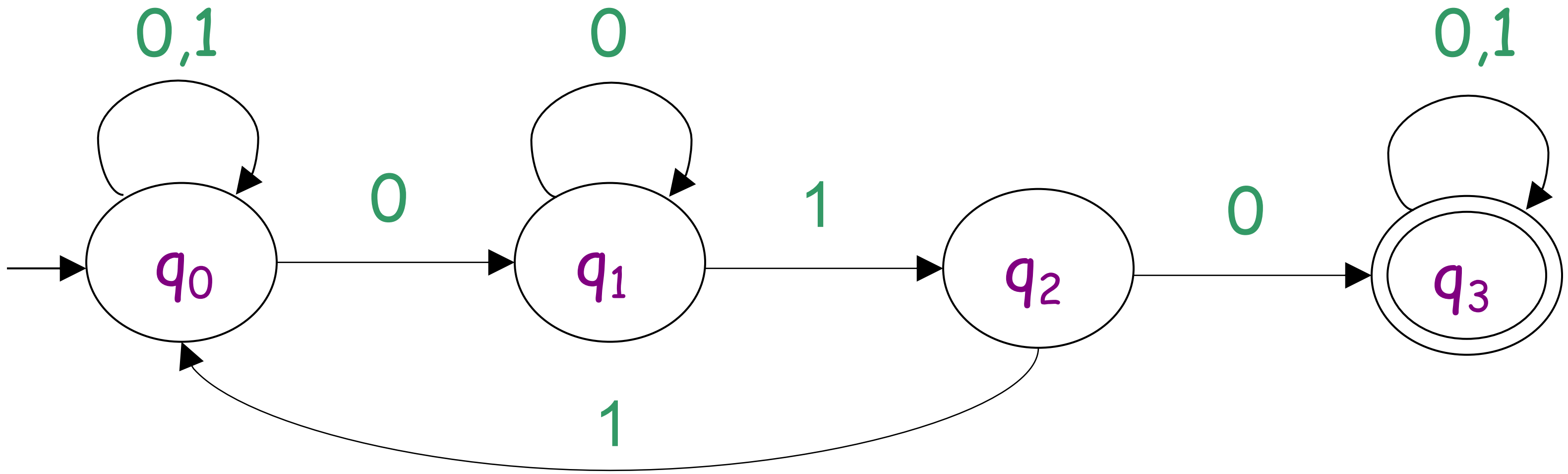
t-DBW

HD-NBW for L with state space Q $\longrightarrow$ DBW for L with state space 2^{Q}

$$\delta'(S,\sigma) = \begin{cases} \delta(S,\sigma) & \text{If } \delta(S,\sigma) \cap \alpha = \emptyset \\ \delta(S,\sigma) \cap \alpha & \text{Otherwise} \end{cases}$$

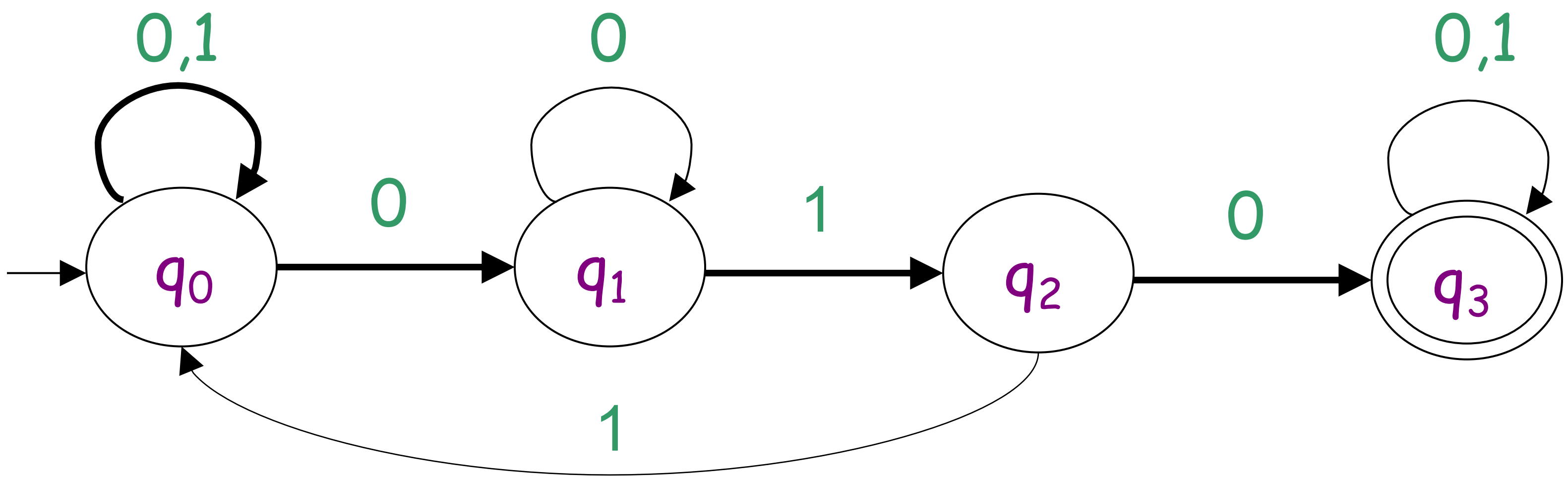
α-transition

Example: An HD-NFW



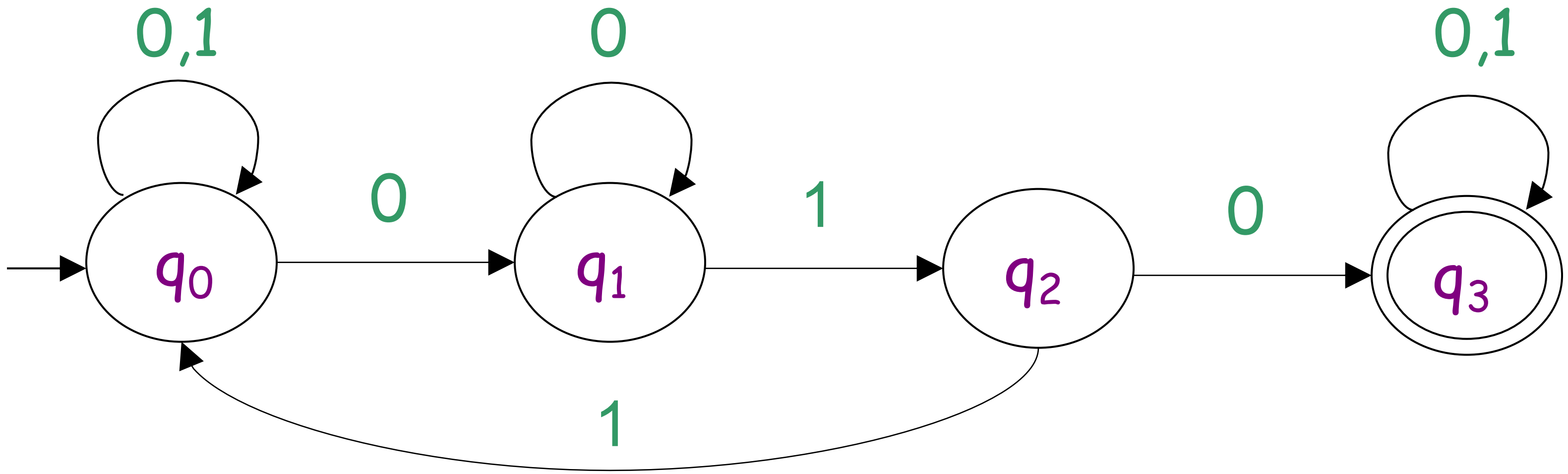
L(A) = {w : w contains 010}

Example: An HD-NFW



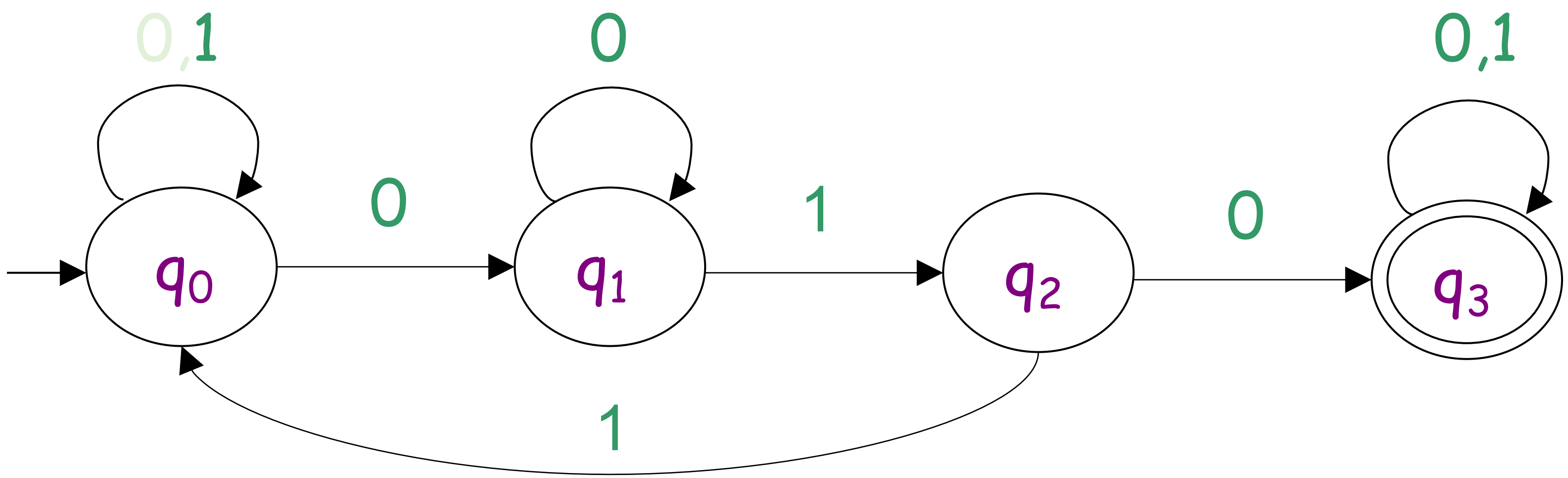
L(A) = {w : w contains 010}

Example: An HD-NFW



L(A) = {w : w contains 010}

Example: An HD-NFW



L(A) = {w : w contains 010}

DBP: determinizable by pruning

Every DBP automaton is HD:
$f(w)=\delta(f(\varepsilon),w)$

A related notion: Semantic Determinism (SD)

Intuitively, nondeterministic choices lead to states with the same language.

Formally, $A=\langle\Sigma,Q,Q_0,\delta,\alpha\rangle$ is HD if for all states $q\in Q$ and letters $\sigma\in\Sigma$, all the states in $\delta(q,\sigma)$ have the same language.

Note: every HD automaton can be pruned to an equivalent SD automaton.

If $\delta(q,\sigma)=\{q_1,q_2,q_3\}$ and $L(A^{q2}) \subsetneq L(A^{q3})$, then $\delta(q,\sigma):=\{q_1,q_3\}$

A related notion: Semantic Determinism (SD)

Intuitively, nondeterministic choices lead to states with the same language.

Formally, $A=\langle \Sigma,Q,Q_0,\delta,\alpha \rangle$ is HD if for all states $q \in Q$ and letters $\sigma \in \Sigma$, all the states in $\delta(q,\sigma)$ have the same language.

Note: every HD automaton can be pruned to an equivalent SD automaton.

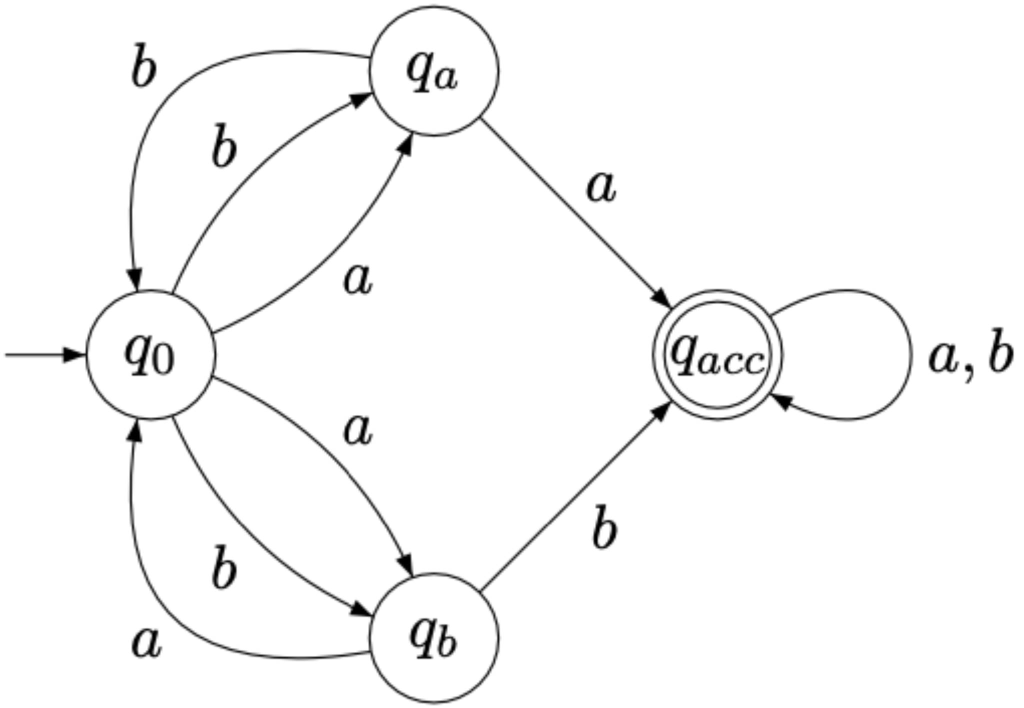
in polynomial time

Are there SD automata that are not HD?

Finite words: No (all SD-NFWs are DBP)

Proof: Easy

Infinite words: Yes!



For all states q, $L(A^q)=(a+b)^\omega$

[Abu-Radi, Kupferman, Leshkowitz 2021]

Are there HD automata that are not DBP?

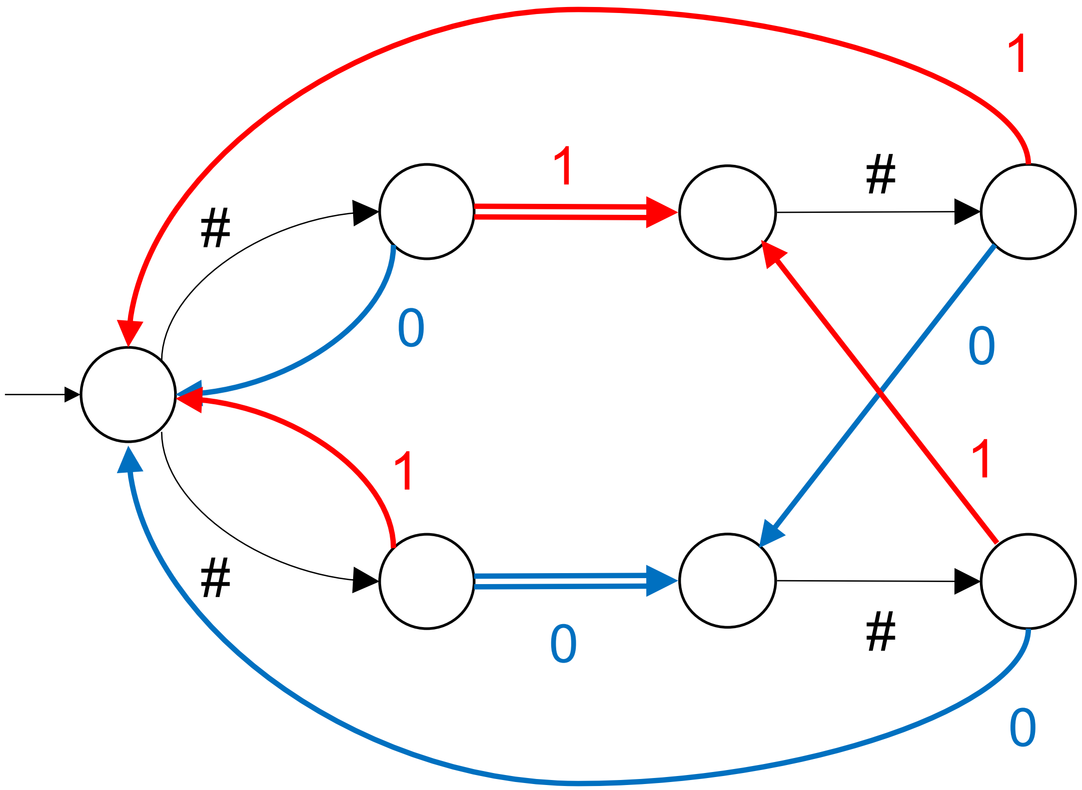
HP06: NBW $\longrightarrow$ HD-NPW, which is DBP

Finite words: No (all HD-NFWs are DBP)

Proof: Easy

Infinite words: Yes!

BKKS 2013: HD automata need not be DBP!



A is a tNBW

L(A)⊆$(\#0+\#1)^{\omega}$

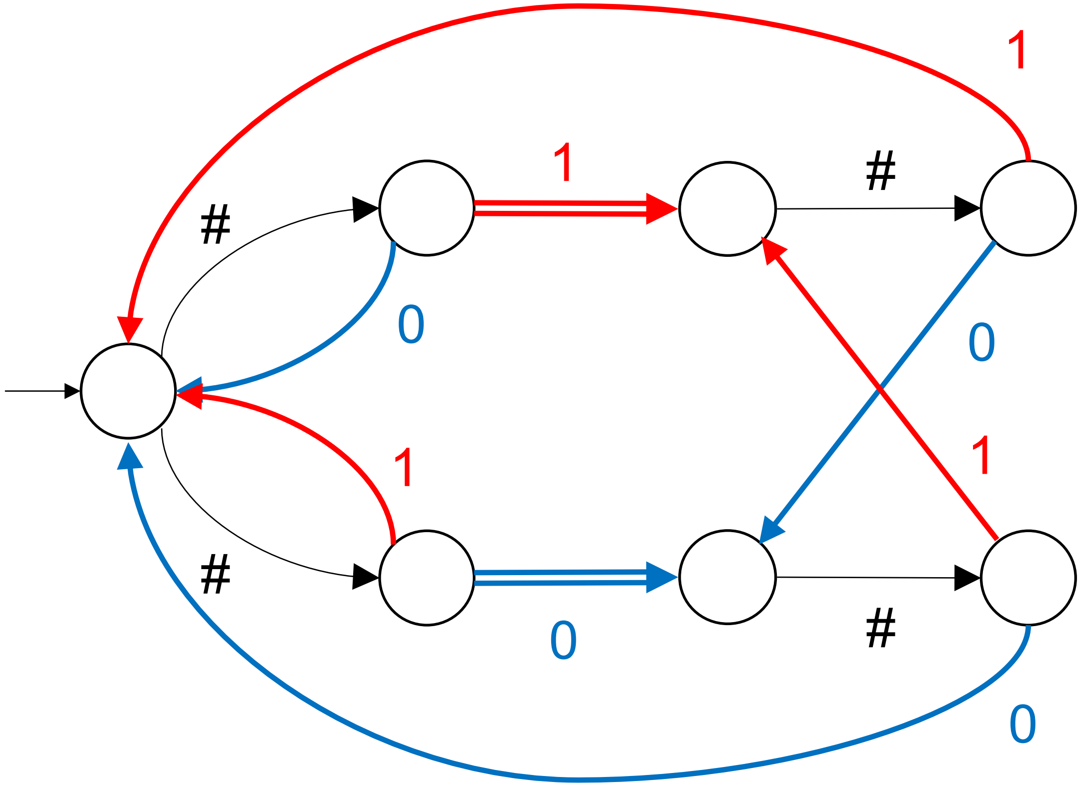
traversing an α-transition

↕

reading #0#0 or #1#1

L(A)= {w∈$(\#0+\#1)^{\omega}$: w has infinitely many #0#0 or #1#1 }

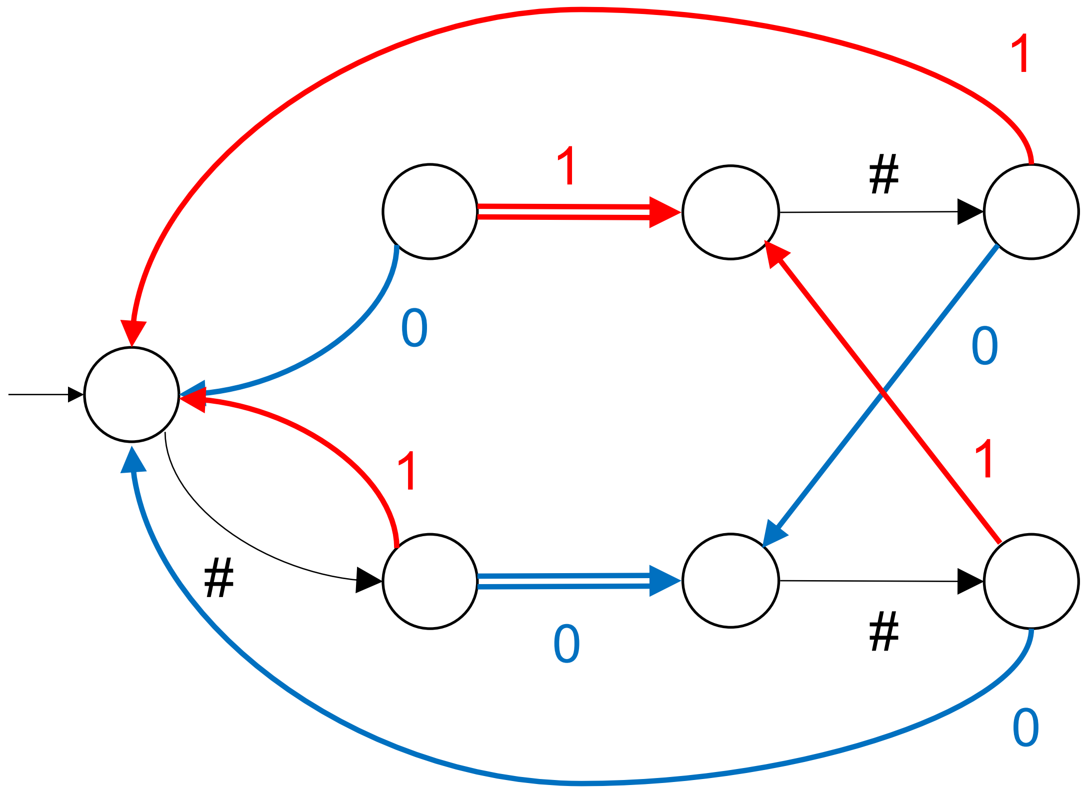
BKKS 2013: HD automata need not be DBP!



Not DBP:

L(A)= {w∈(#0+#1)$^{\omega}$: w has infinitely many #0#0 or #1#1 }

BKKS 2013: HD automata need not be DBP!

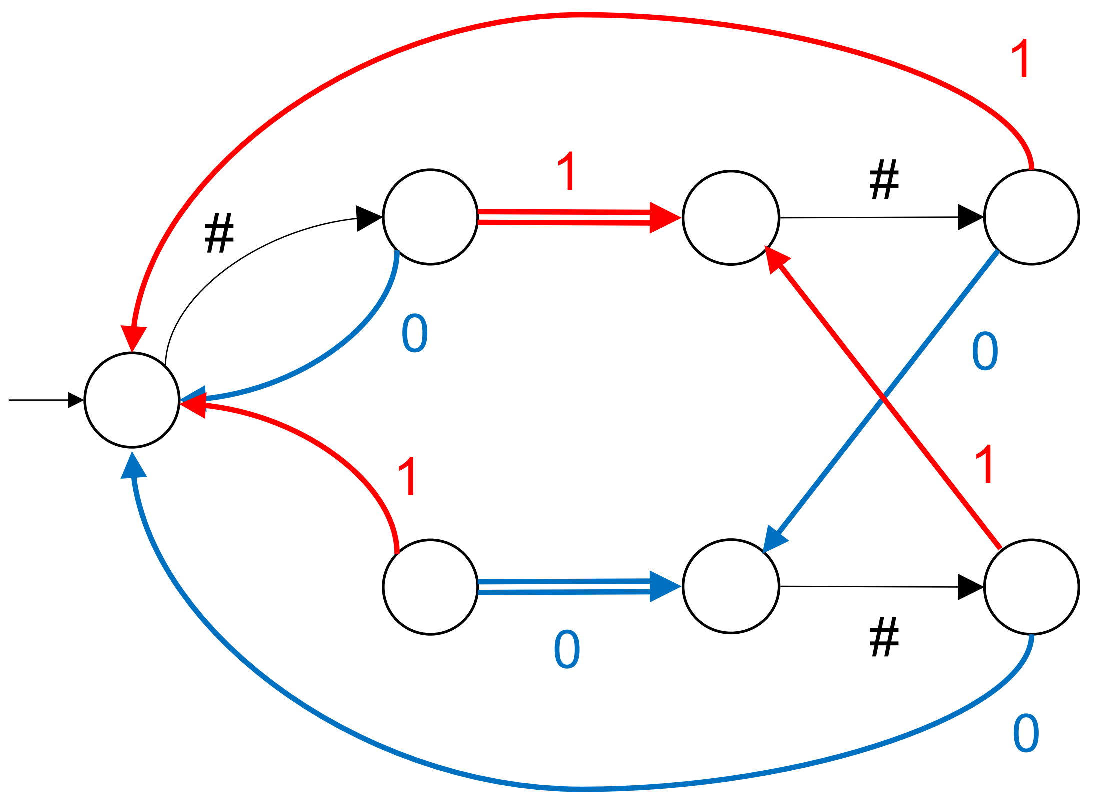


Not DBP:

Pruning 1: rejects $(\#1)^{\omega}$

L(A)= $\{w \in (\#0+\#1)^{\omega}$: w has infinitely many #0#0 or #1#1 $\}$

BKKS 2013: HD automata need not be DBP!



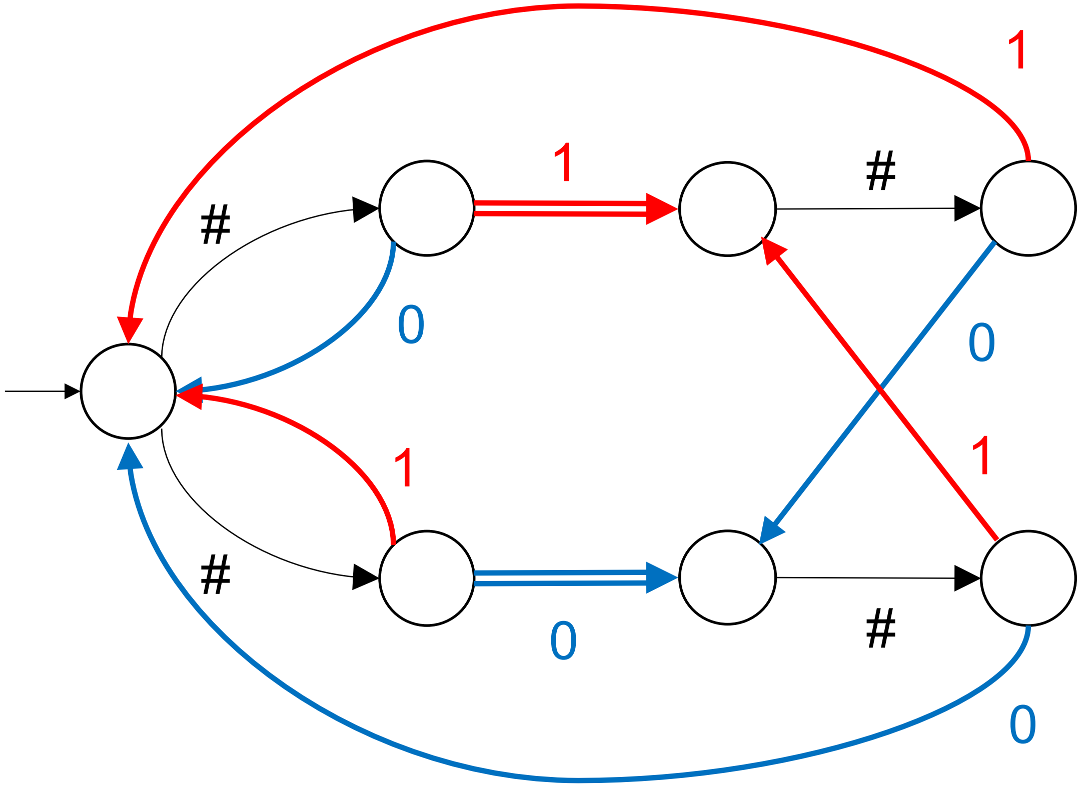
Not DBP:

Pruning 1: rejects $(\#1)^\omega$

Pruning 2: rejects $(\#0)^\omega$

L(A)= $\{w \in (\#0+\#1)^\omega$: w has infinitely many #0#0 or #1#1 $\}$

BKKS 2013: HD automata need not be DBP!



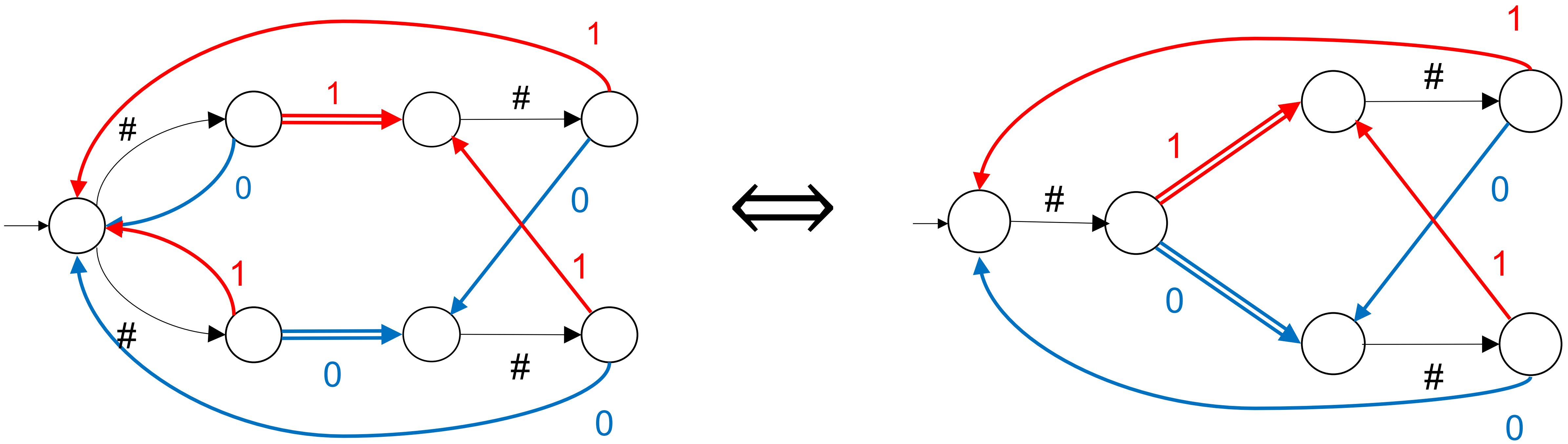
HD strategy in q_0:
- if you arrive with 1, go up
- if you arrive with 0, go down

L(A)= {w∈(#0+#1)$^{\omega}$: w has infinitely many #0#0 or #1#1 }

So, HD automata need not be DBP!

Can HD automata be smaller than deterministic ones?

The previous example does not imply this:



So, HD automata need not be DBP!

Can HD automata be smaller than deterministic ones?

Finite words: No (all HD-NFWs are DBP)

Infinite words:

So, HD automata need not be DBP!

Can HD automata be smaller than deterministic ones?

Finite words: No (all HD-NFWs are DBP)

Infinite words: Open!

So, HD automata need not be DBP!

Can HD automata be smaller than deterministic ones?

Finite words: No (all HD-NFWs are DBP)

Infinite words: Open!

Kuperberg, Skrzypczak 2016:

Kuperberg, Skrzypczak 2016:

HD-NCW are exponentially more succinct than DCW! 😱😱

NCW: nondeterministic co-Büchi word automata

A run is accepting if it visits states in α only finitely often

Kuperberg, Skrzypczak 2016:

HD-NCW are exponentially more succinct than DCW! 😱😱

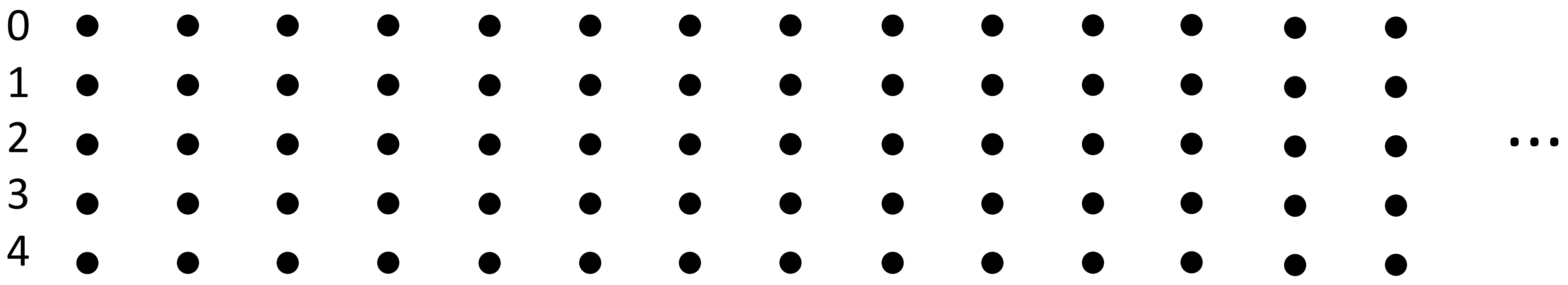
A family of languages L_1,L_2,L_3,... such that for every n≥1:

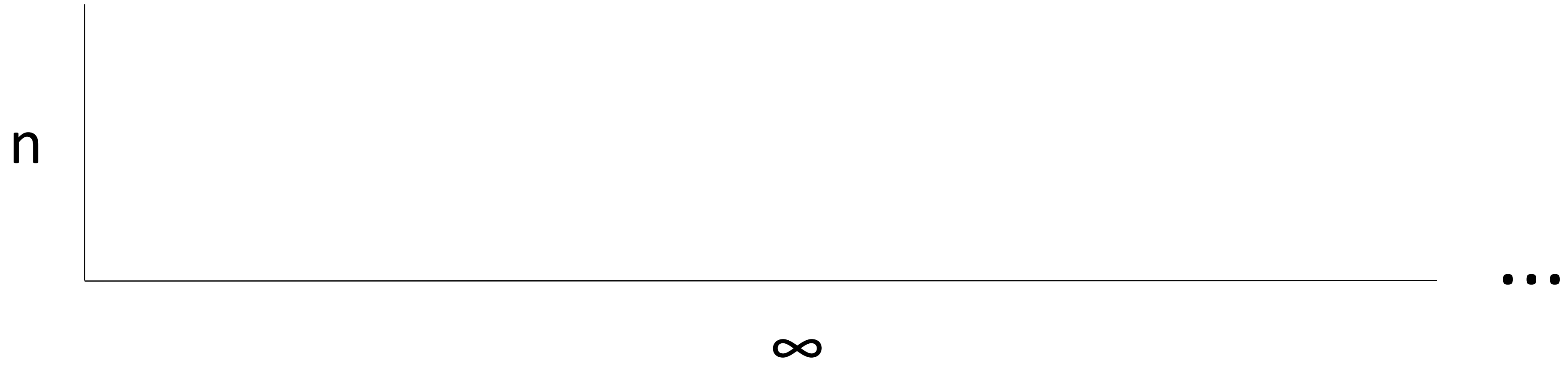
- L_n can be recognized by a HD-NCW with n states
- A minimal DCW for L_n needs 2^n states

NCW: nondeterministic co-Büchi word automata

A run is accepting if it visits states in α only finitely often

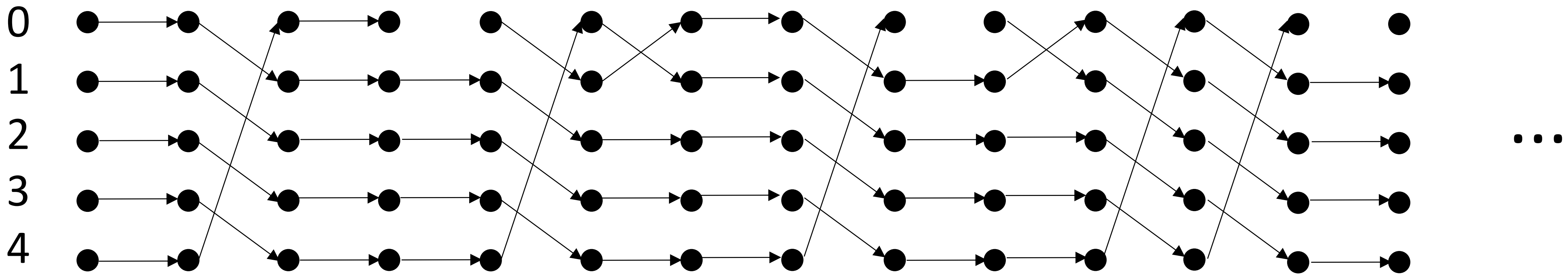
The language L_n: n=5

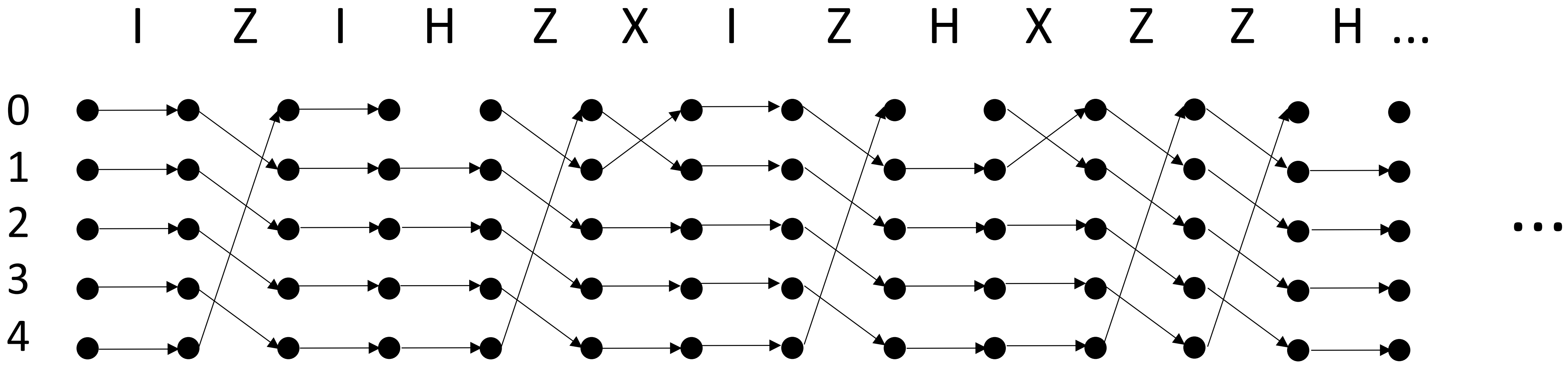




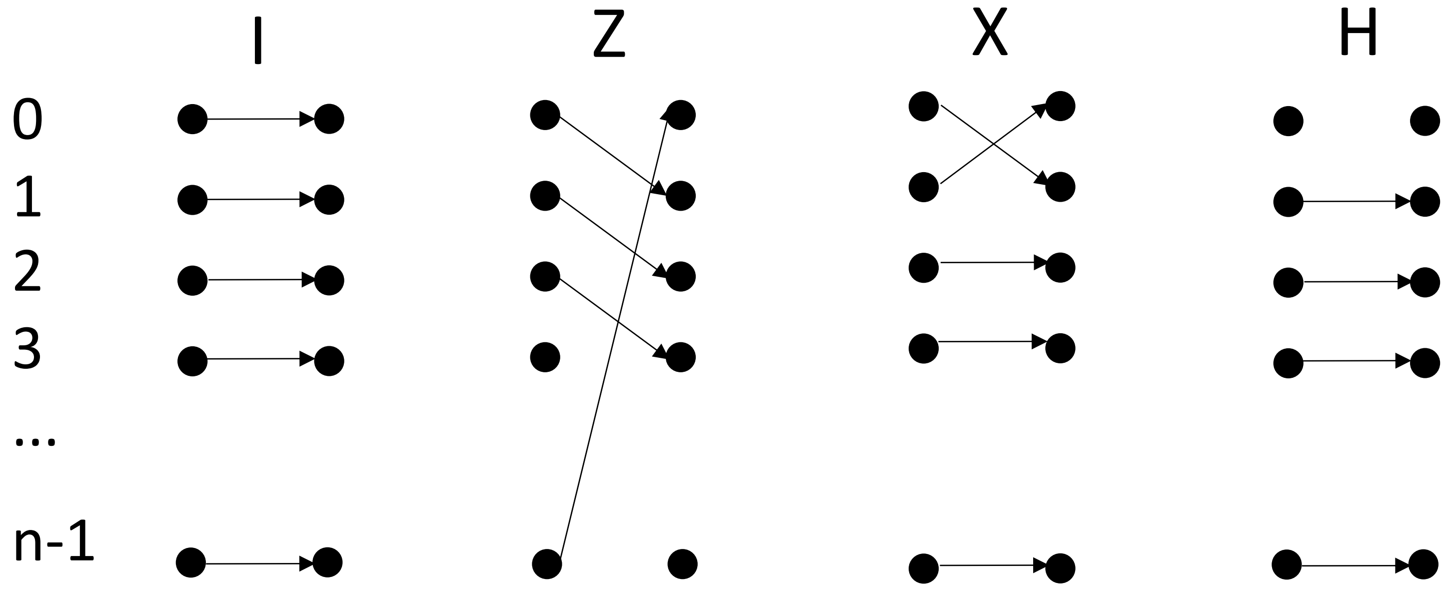
The language L_n:

n=5

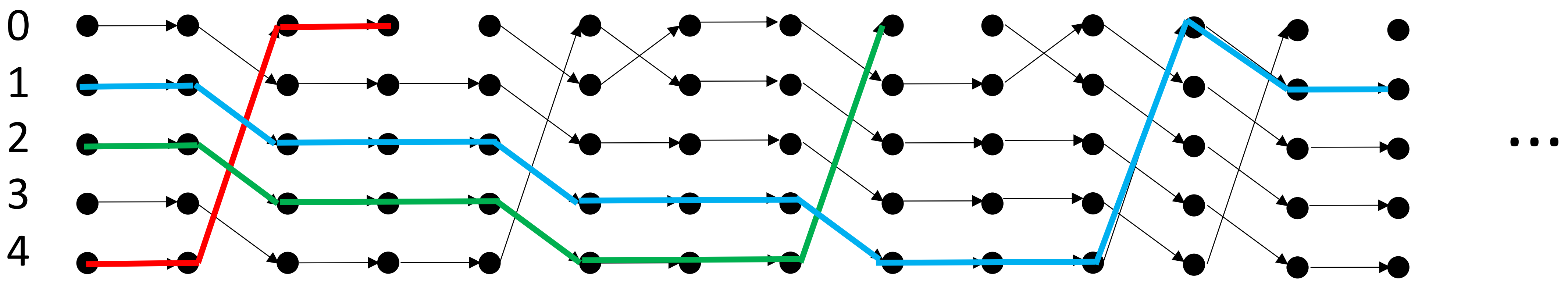




Four letters:



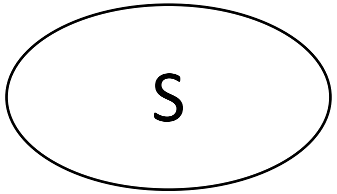
The language L_n:



$\Sigma = \{I,Z,X,H\}$

L_n = {w: the graph induced by w contains an infinite line}

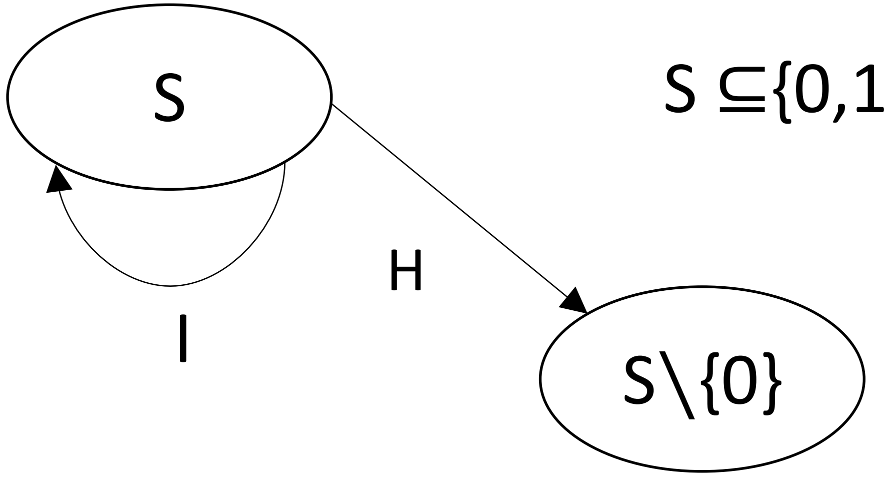
A DCW for L_n:



$S \subseteq \{0,1,2,\ldots,n-1\}$

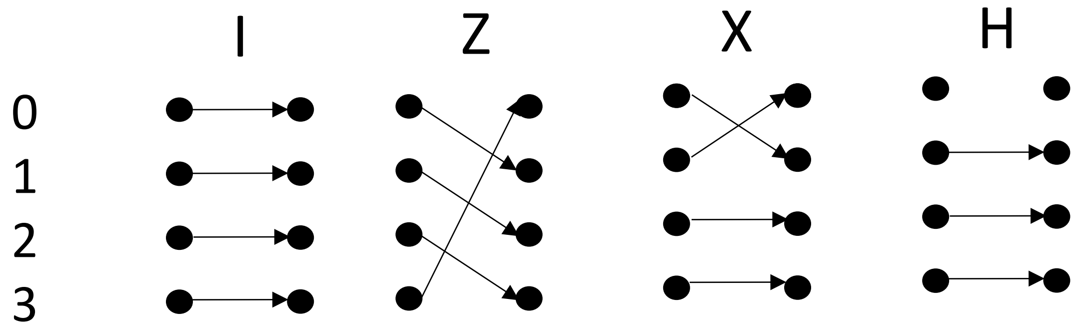
S: the set of lines that survive since the previous break-point

A DCW for L_n:

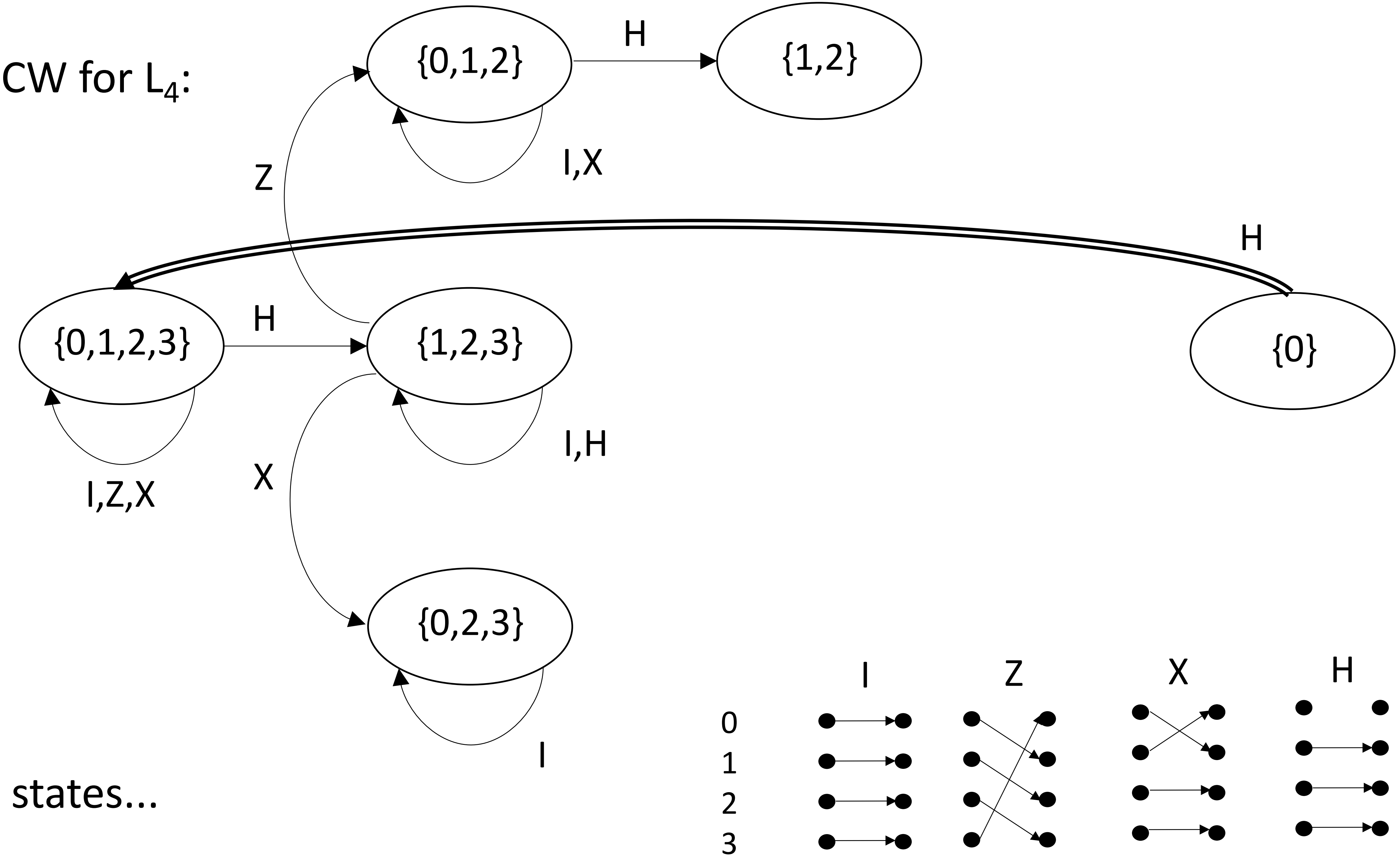


$S \subseteq \{0,1,2,\ldots,n-1\}$

S: the set of lines that survive since the previous break-point



A tDCW for L_4:

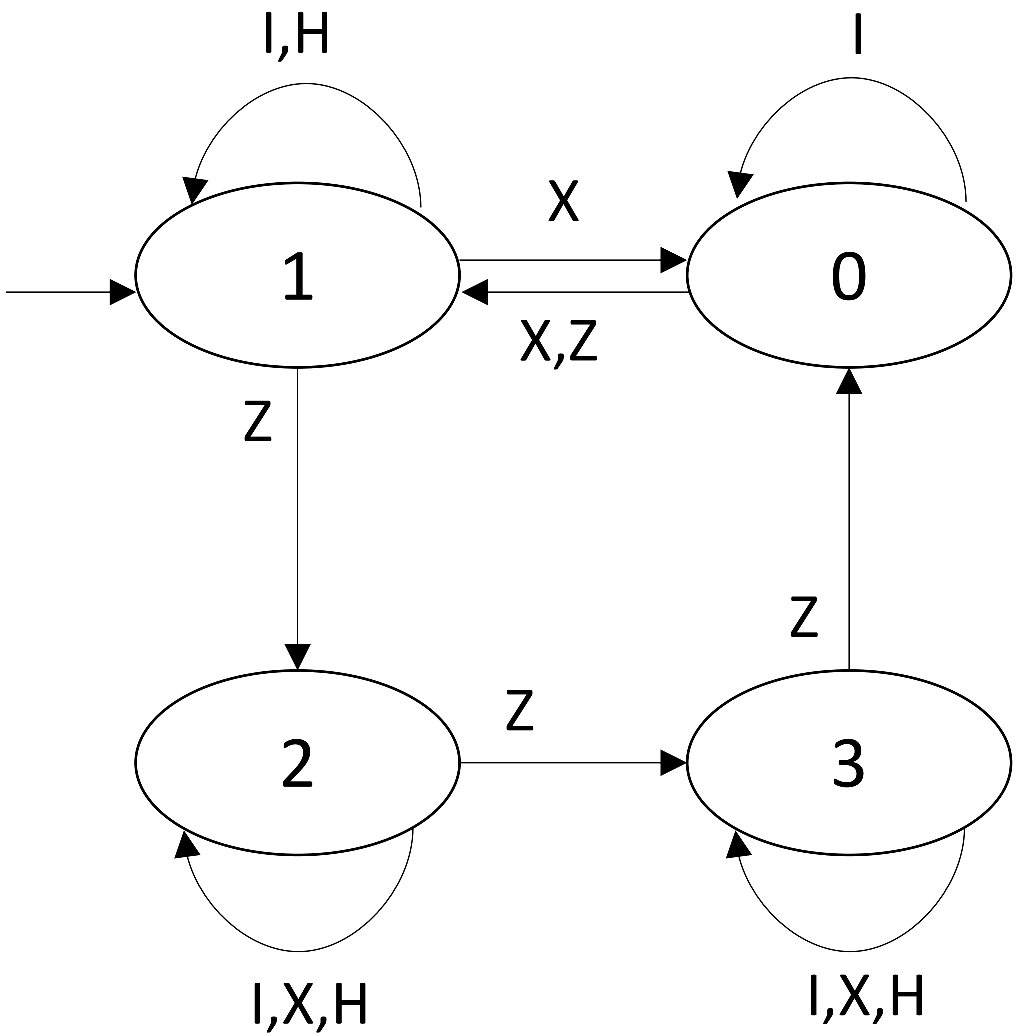


2^n-1 states...

A tNCW for L_n: guesses a line that survives

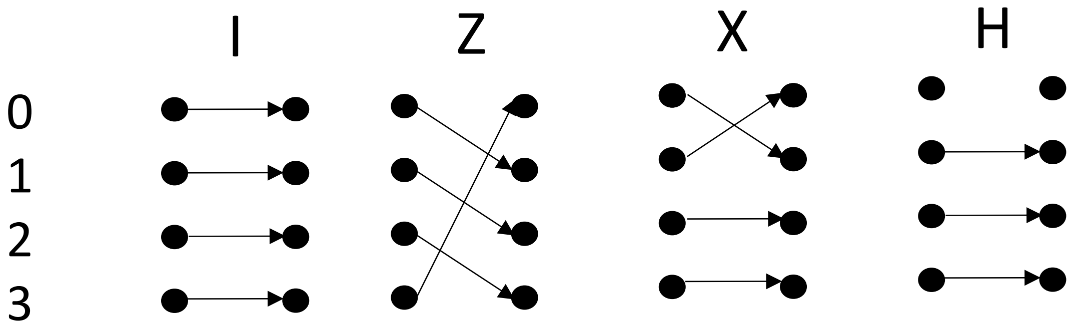
$Q = \{0,1,2,...,n-1\}$

A tNCW for L_n: guesses a line that survives

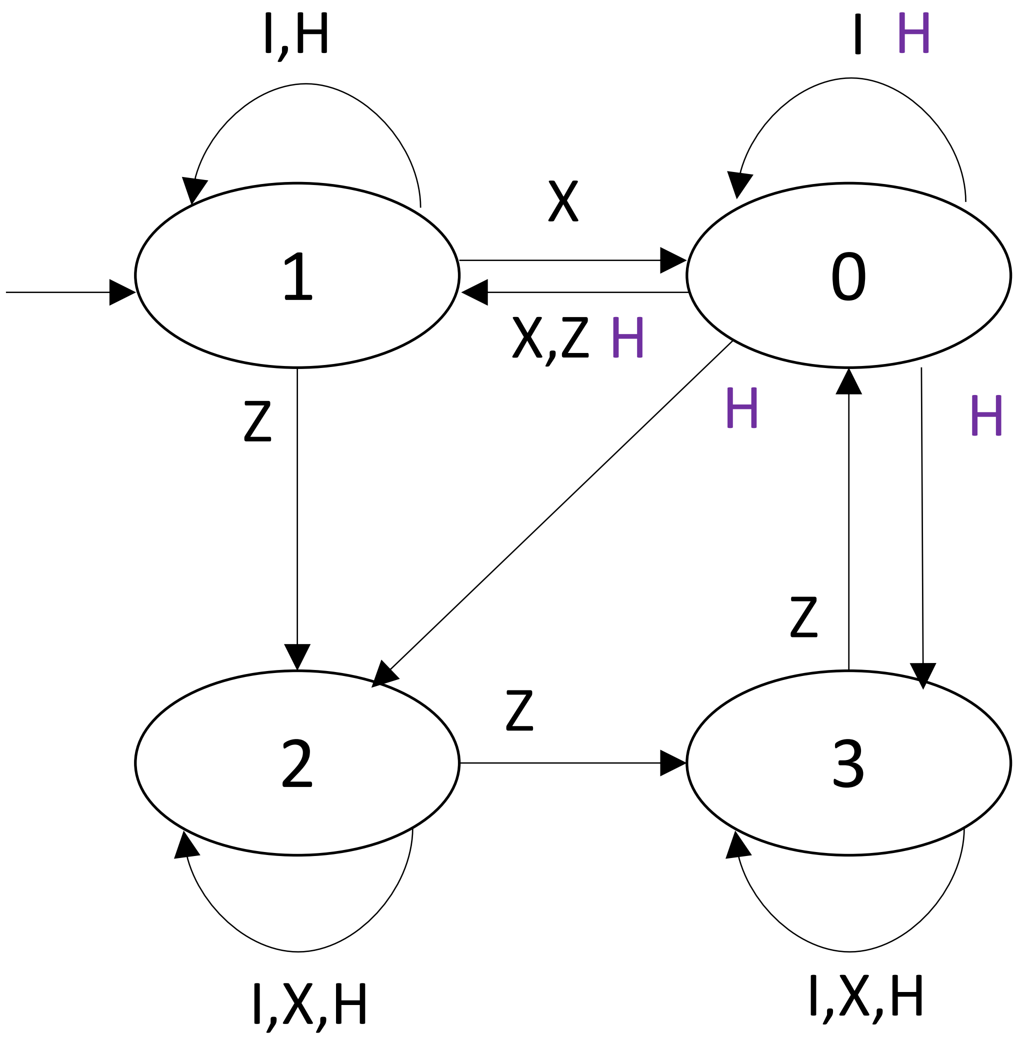


$Q = \{0,1,2,\ldots,n-1\}$

If you follow line 0 and read H, take an α-transition to a new line



A tNCW for L_n: guesses a line that survives

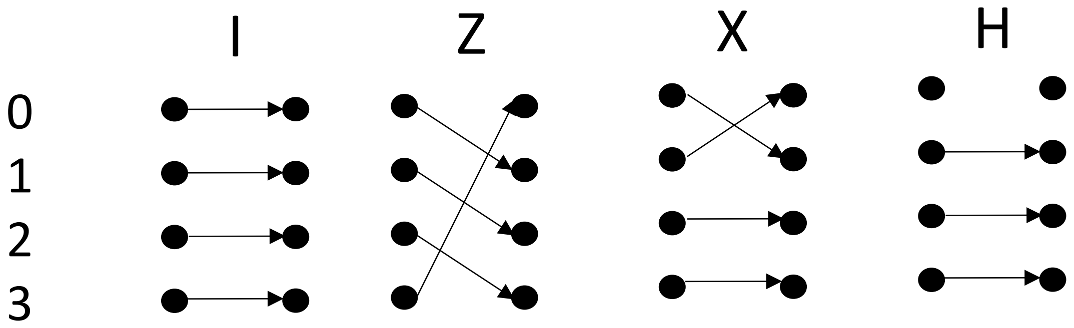


$Q = \{0,1,2,\ldots,n-1\}$

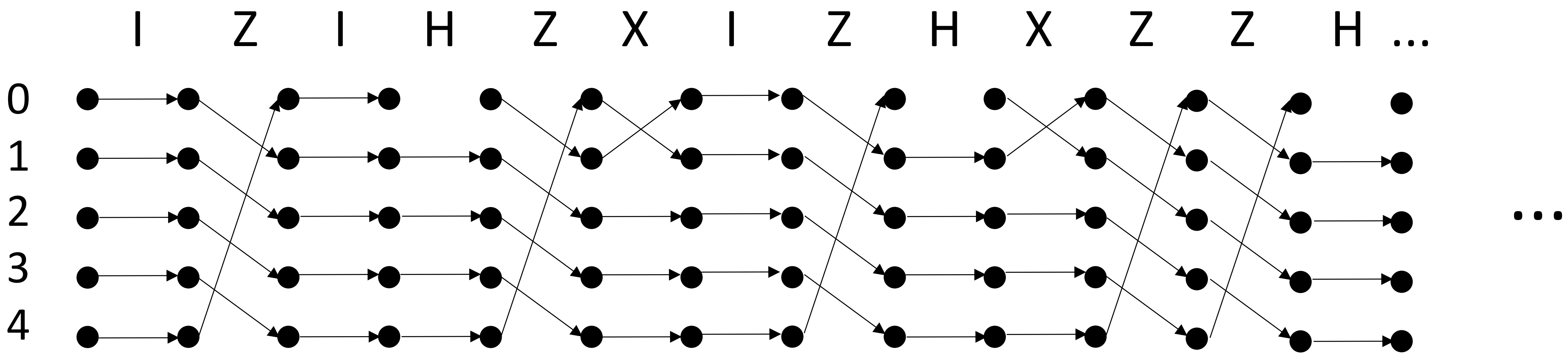
If you follow line 0 and read H,
take an α-transition to a new line

Easy: correct

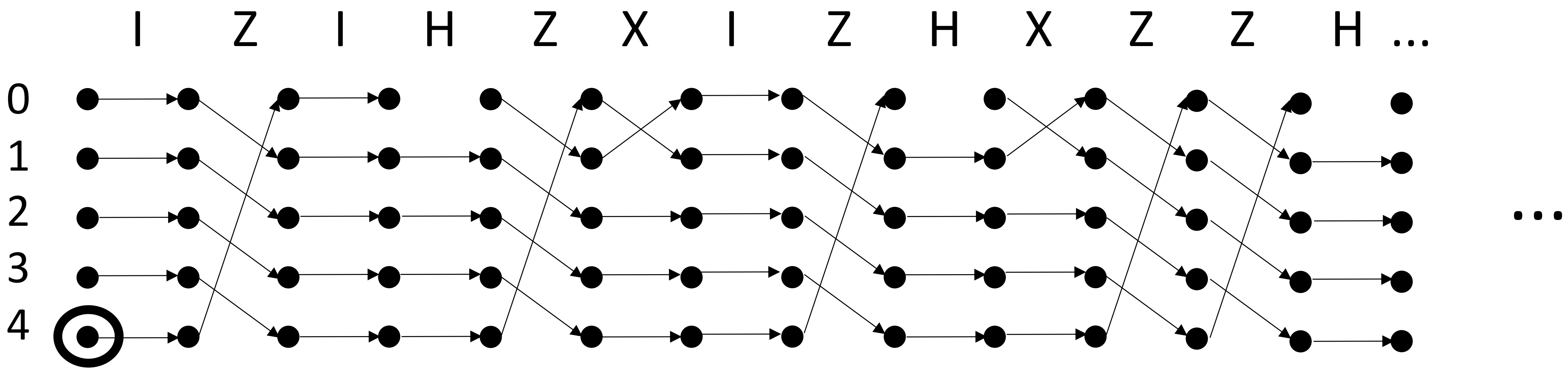
Hard: HD



A HD strategy:

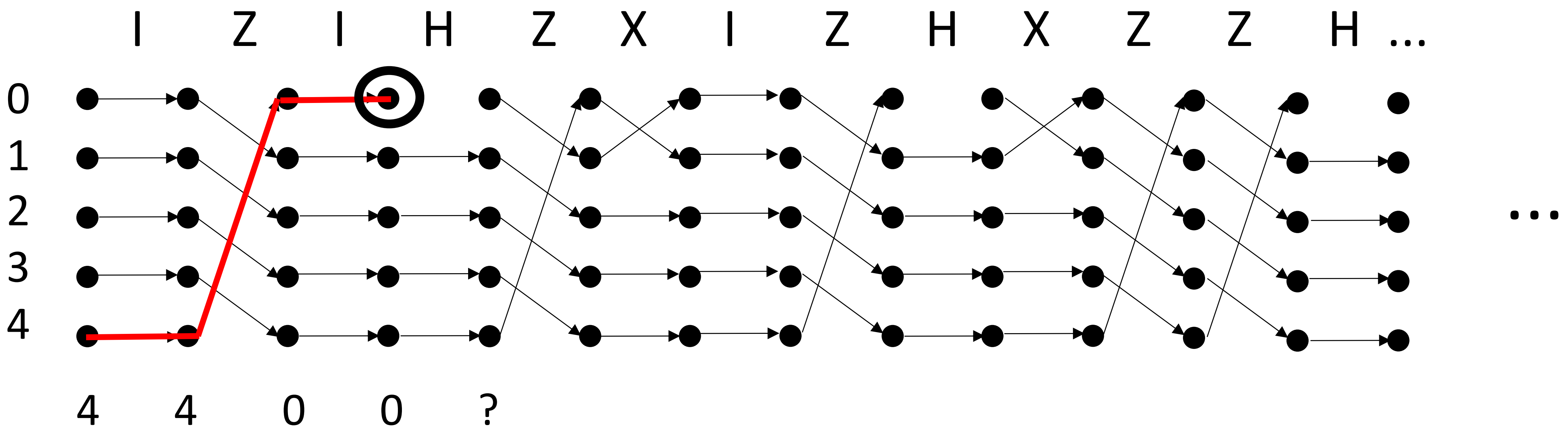


An HD strategy:



- Suppose we start with 4..

An HD strategy:



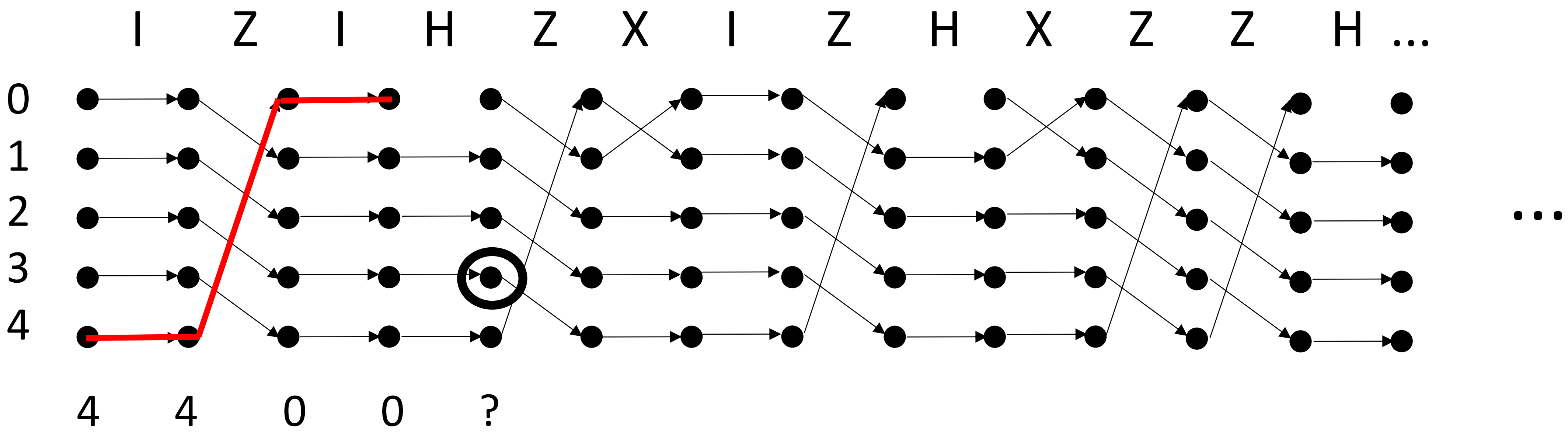
- Suppose we start with 4..

- Where to go?

- HD strategy: To a line with a longest history!

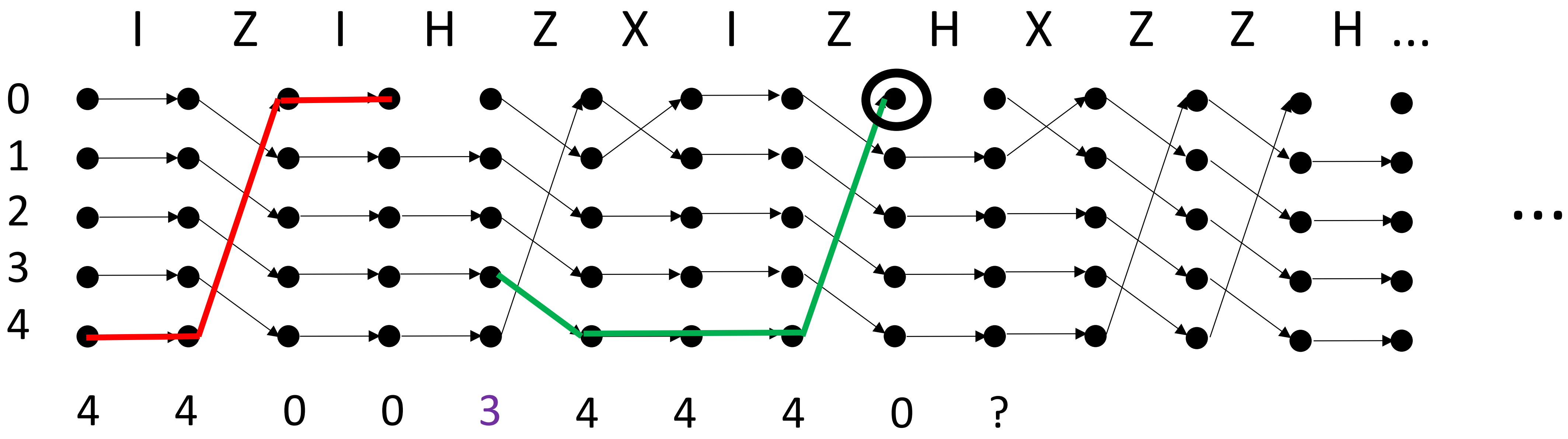
- The idea: If an infinite line exists, it would eventually become a line with a longest history!

An HD strategy:



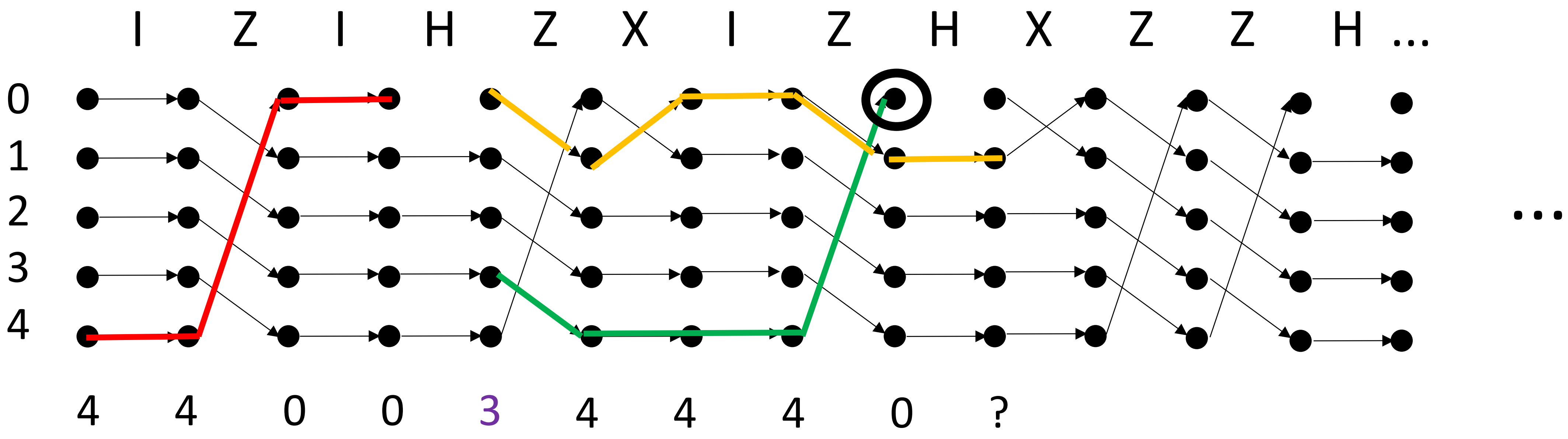
- Suppose we start with 4..
- Where to go?
- Suppose we continue with 3...

An HD strategy:



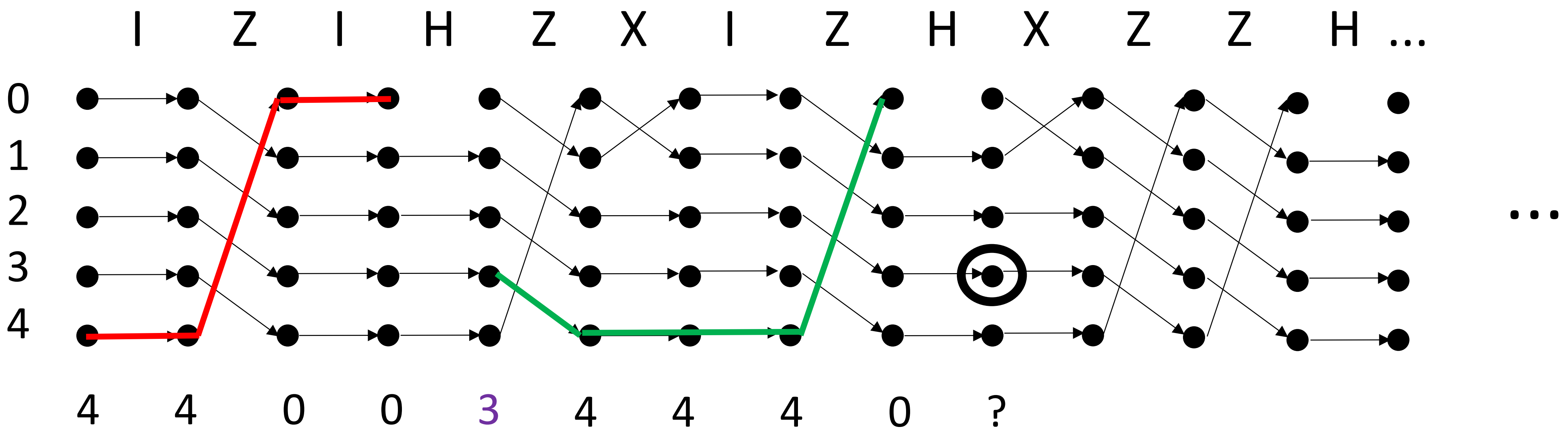
- Suppose we start with 4..
- Where to go?
- Suppose we continue with 3...
- Where to go?

An HD strategy:



- Suppose we start with 4..
- Where to go?
- Suppose we continue with 3...
- Where to go? - Not to 1

An HD strategy:

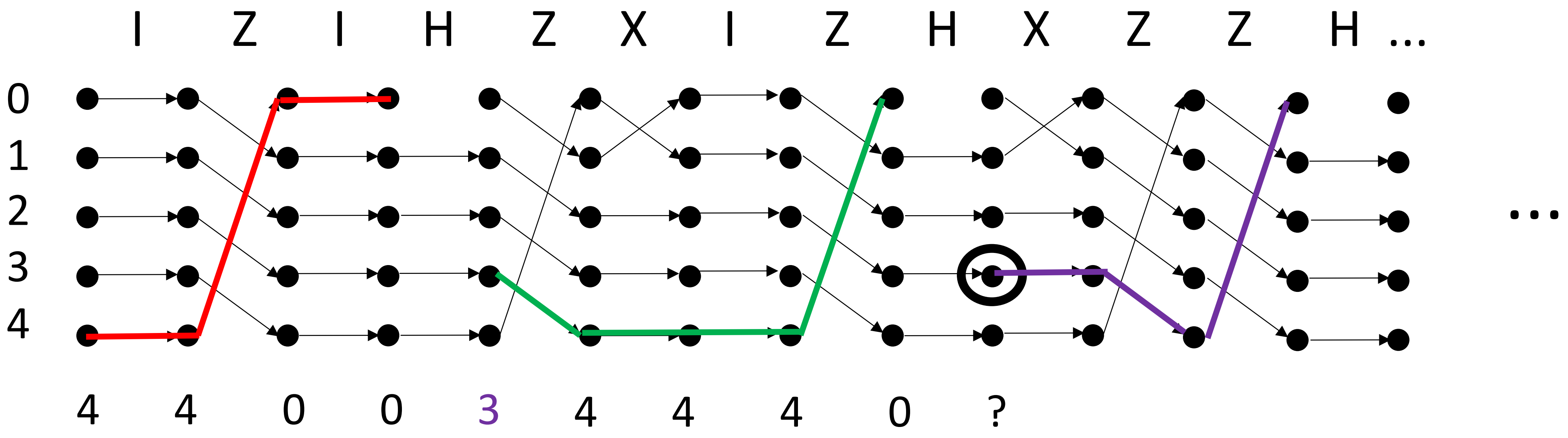


- Suppose we start with 4..

- Where to go?

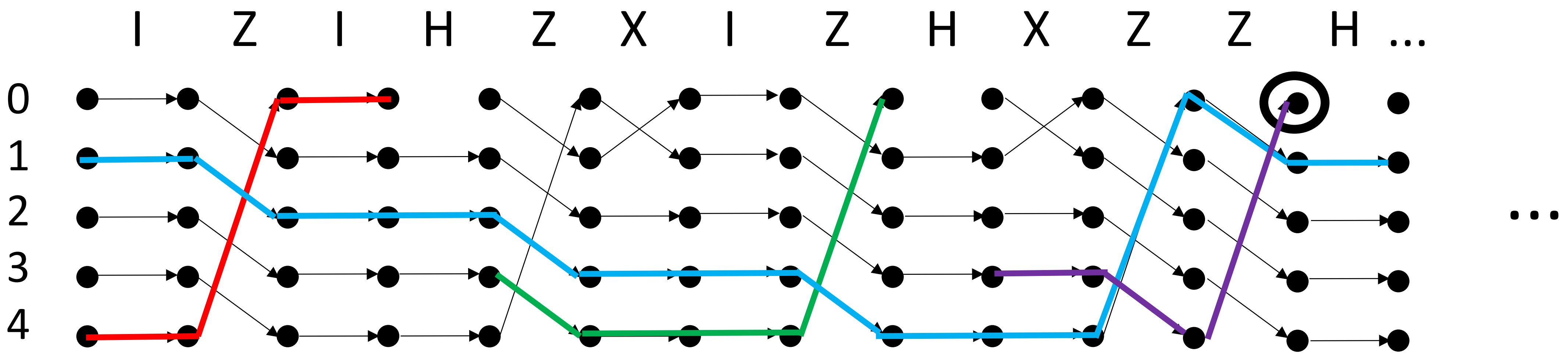
- Suppose we continue with 3...

An HD strategy:



- Suppose we start with 4..
- Where to go?
- Suppose we continue with 3...
- Where to go? - Possibly to 1

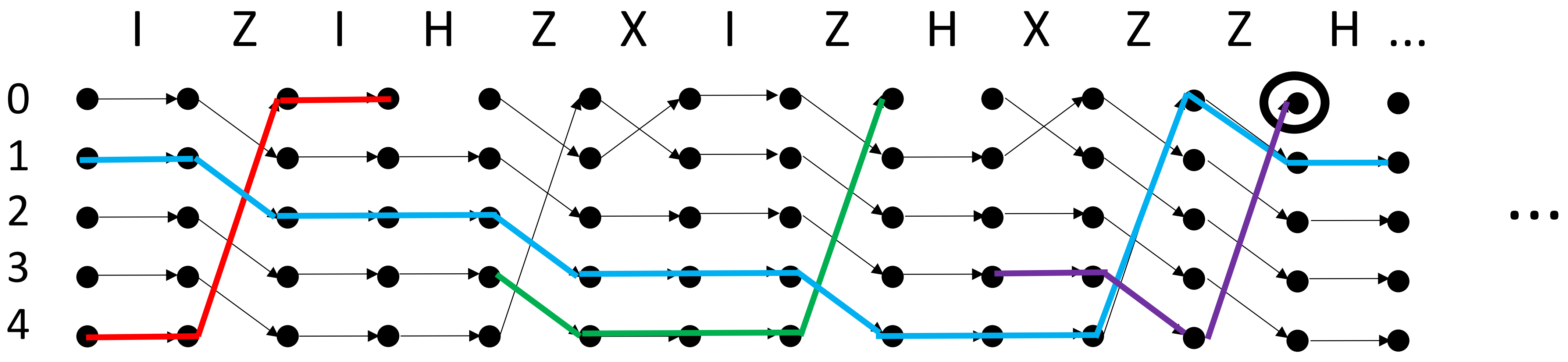
An HD strategy:



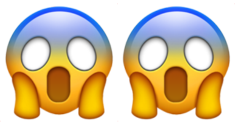
- Suppose we start with 4..
- Where to go?
- To a line with the longest history!
- Suppose we continue with 3..
- Where to go? - Possibly to 1

An HD strategy:

Kuperberg, Skrzypczak 2016



- Indeed, if an infinite line exists, it would eventually become a line with a longest history!

- The past directed us how to resolve nondeterminism! 

- Exponential saving in co-Büchi automata

- Büchi: open!

Many more open HD problems:

- Minimization of HD automata (NP-complete for DBW and DCW, polynomial for HD-tNCW)

- (I/O)-aware HD automata (HD in the input component of the alphabet, nondeterministic in the output component)

- Richer models (Alternating, pushdown, weighted... HD-automata)

- Good-for-X automata (other applications in which current methods use deterministic automata, e.g., probabilistic reasoning)

- Additional models of weak-determinism (DBP, semantically deterministic, ...)

Thank you!