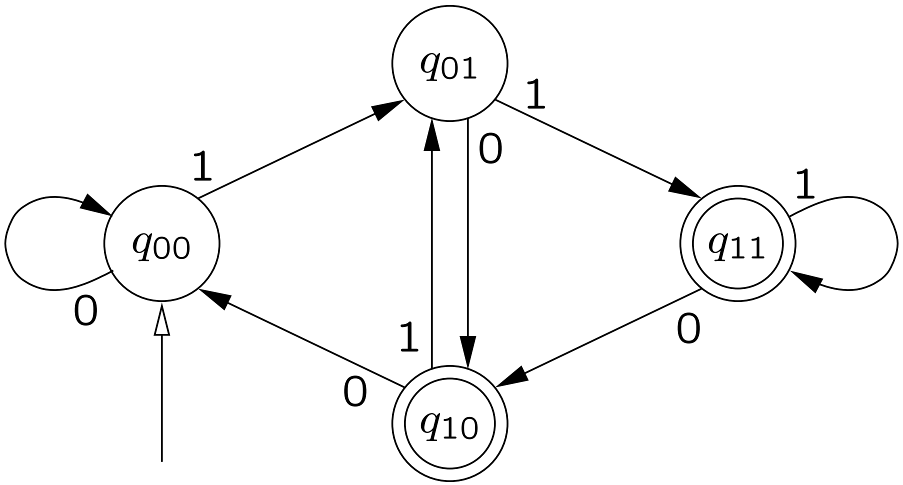
small sweeping 2NFAs are not closed under complement

Christos Kapoutsis

international colloquium on
Automata, Languages and Programming
Venice, Italy, July 2006

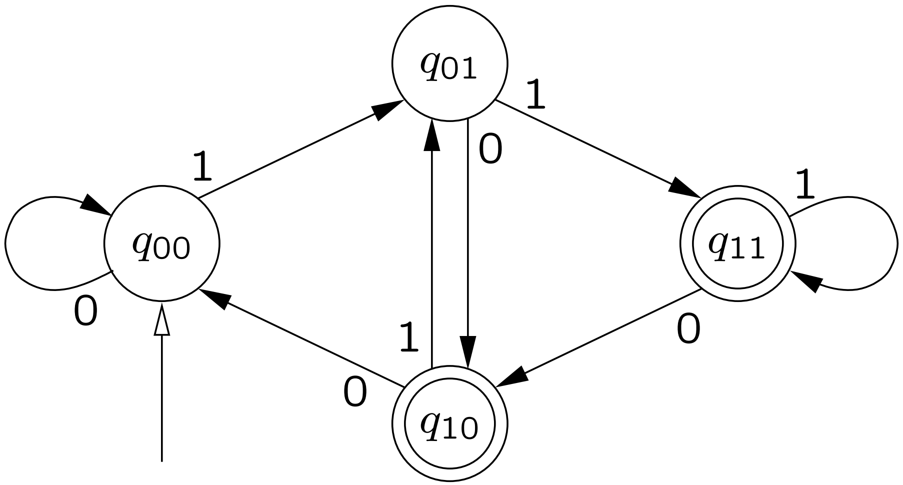
the main problem

DFA

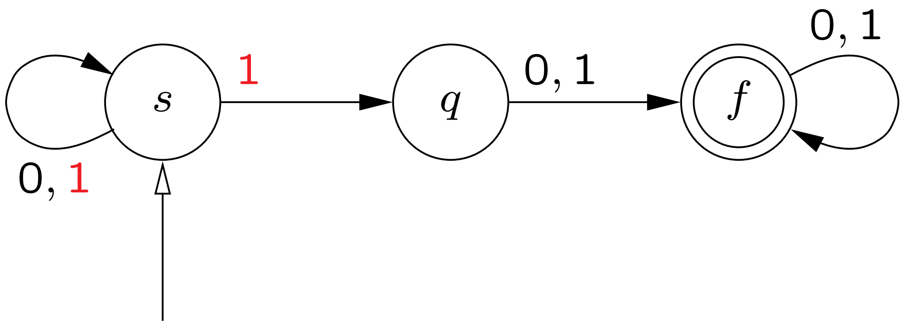


the main problem

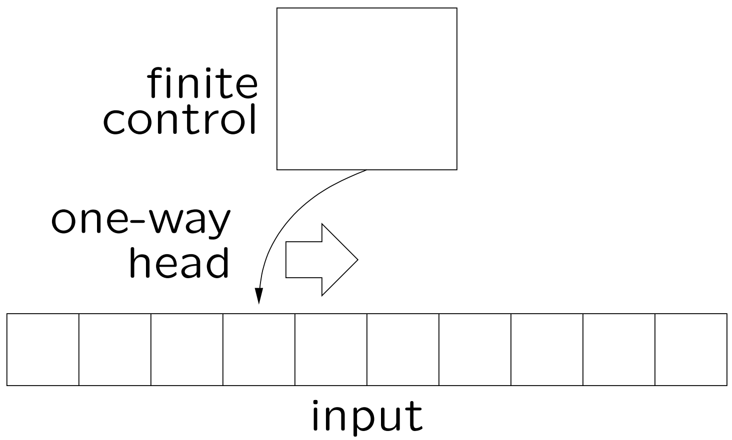
DFA



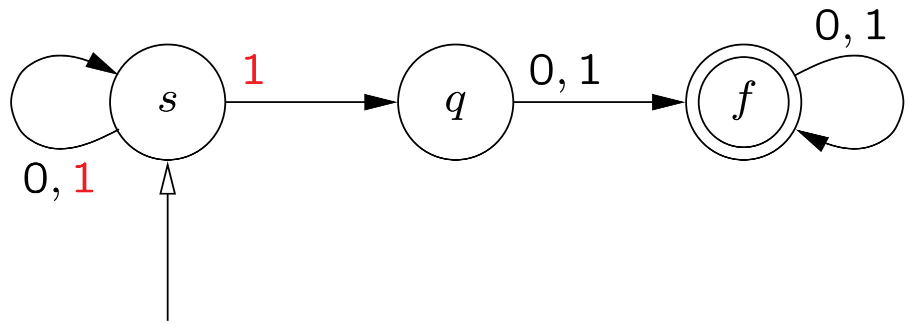
NFA



1DFA

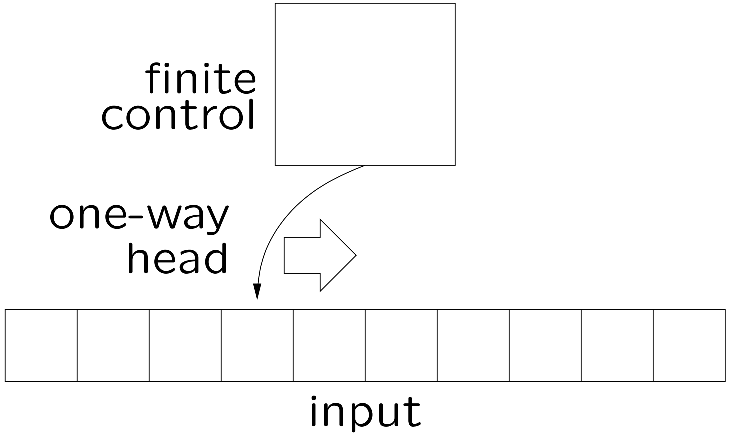


NFA

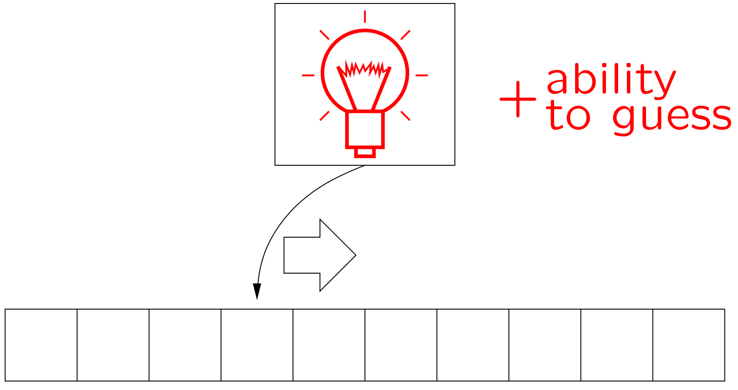


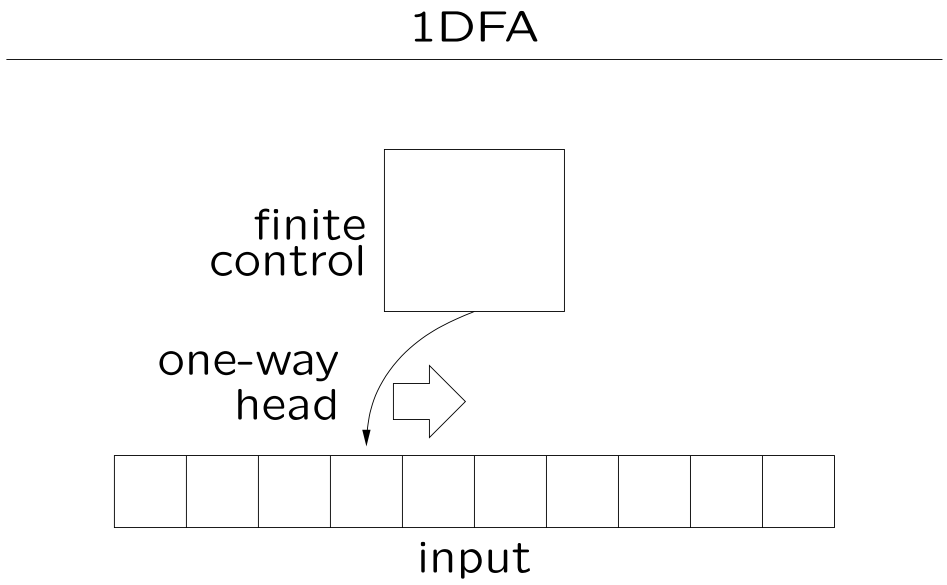
the main problem

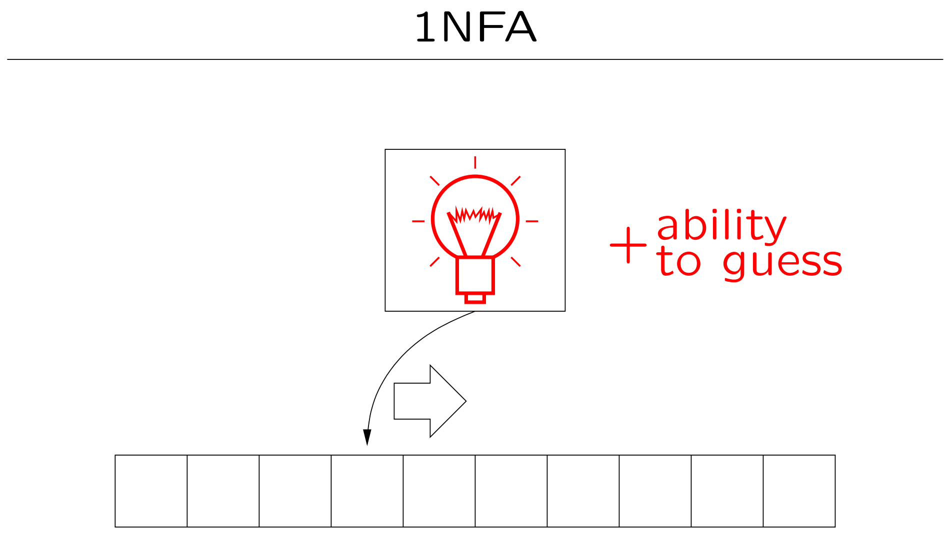
1DFA



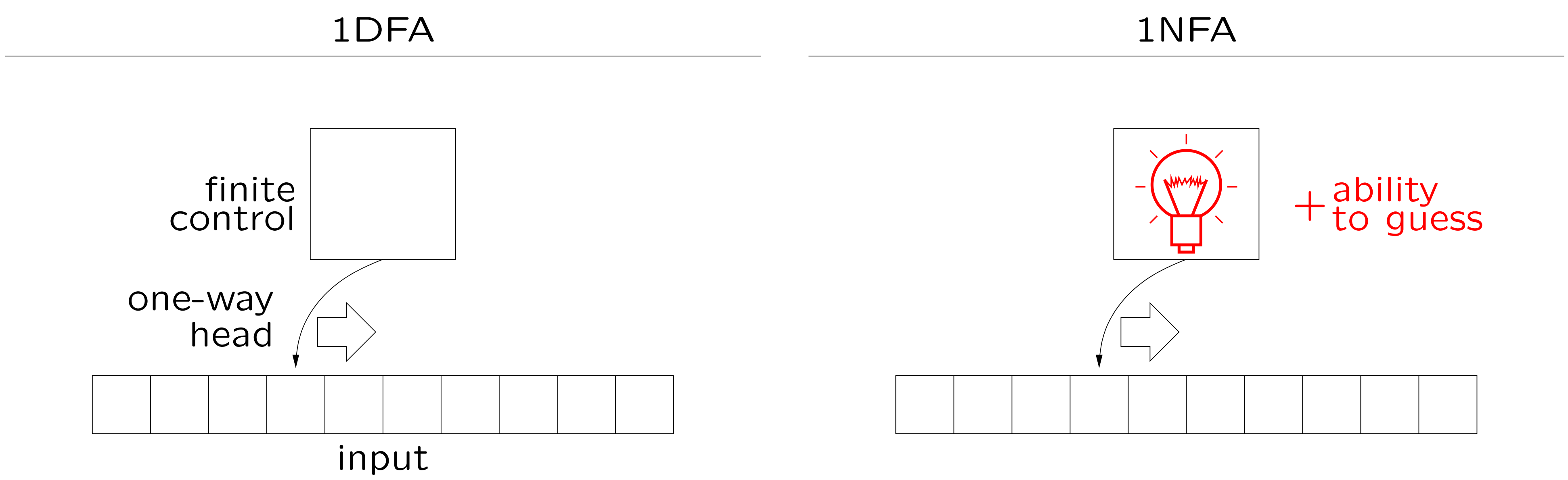
1NFA







a 1DFA with $\leq 2^n - 1$ states ← can be converted to — every 1NFA with n states

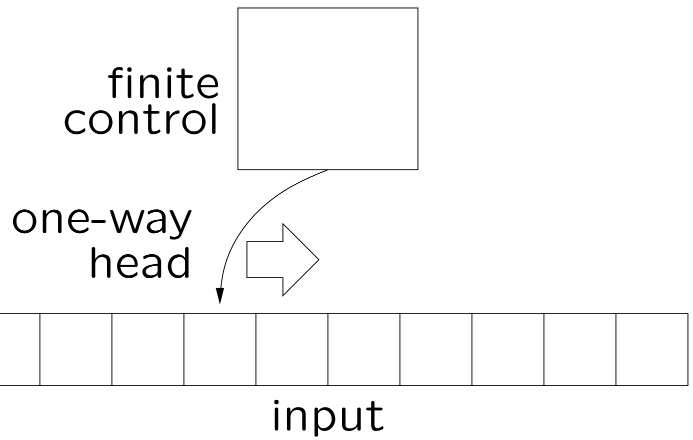


a 1DFA with $\leq 2^n - 1$ states ← can be converted to — every 1NFA with n states

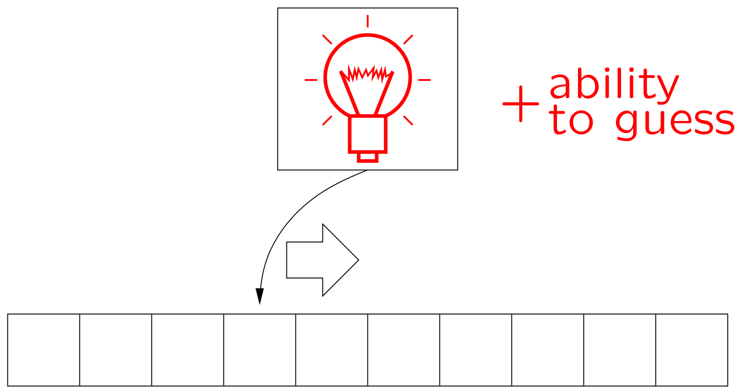
and sometimes all these $2^n - 1$ states are necessary

the main problem

1DFA



1NFA

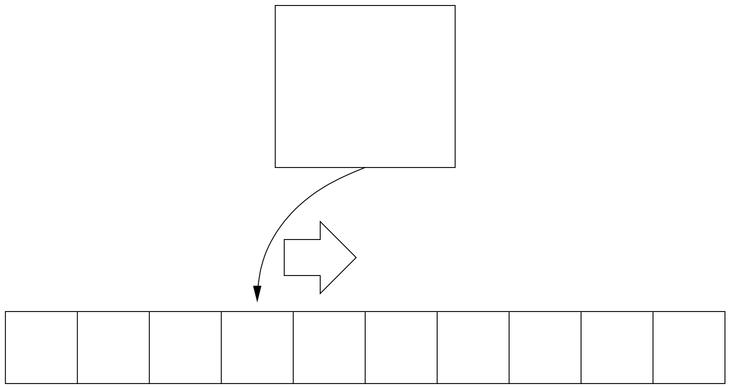
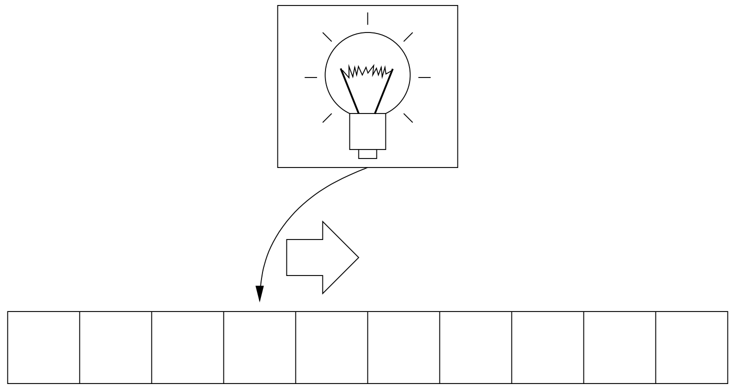


a 1DFA with $\leq 2^n - 1$ states ← can be converted to — every 1NFA with n states

and sometimes all these $2^n - 1$ states are necessary

"*the trade-off is exactly* $2^n - 1$"

the main problem

1DFA	1NFA
	

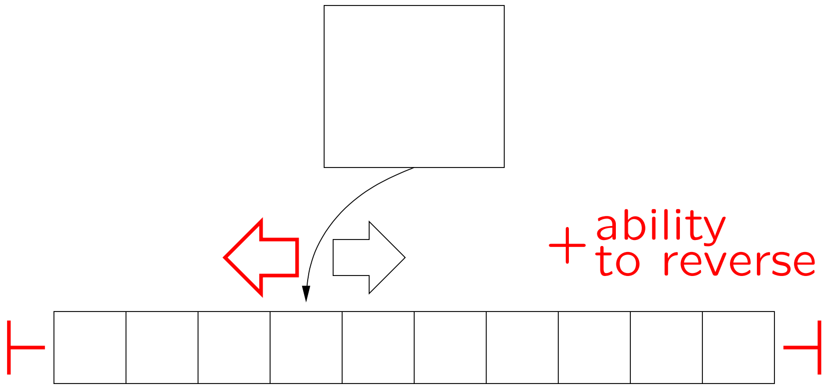
a 1DFA with $\leq 2^n - 1$ states ⟵ *can be converted to* ⟵ every 1NFA with n states

and sometimes all these $2^n - 1$ states are necessary

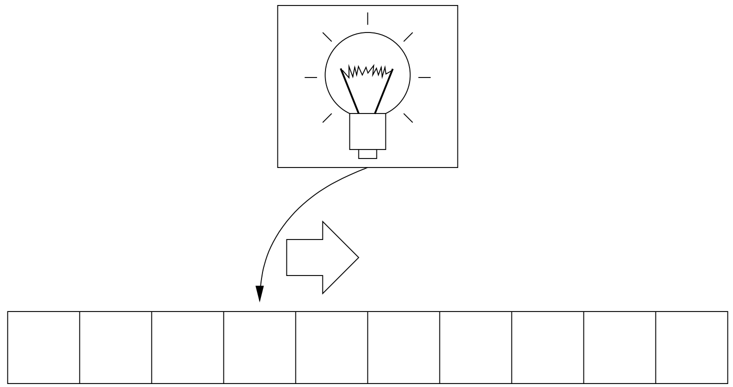
"*the trade-off is exactly* $2^n - 1$"

the main problem

2DFA



1NFA



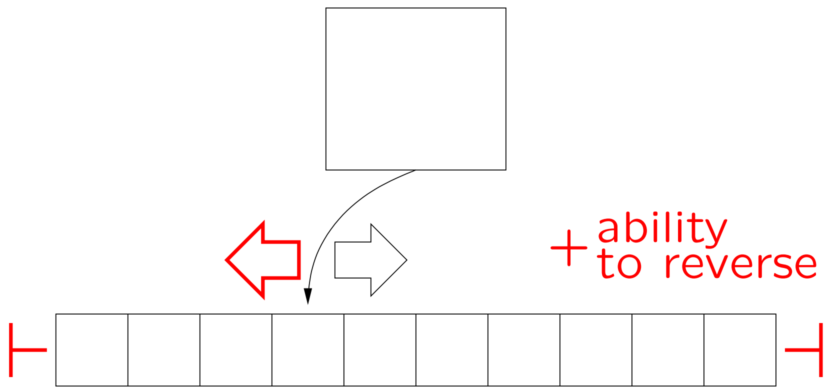
a 1DFA with $\leq 2^n - 1$ states ← can be converted to — every 1NFA with n states

and sometimes all these $2^n - 1$ states are necessary

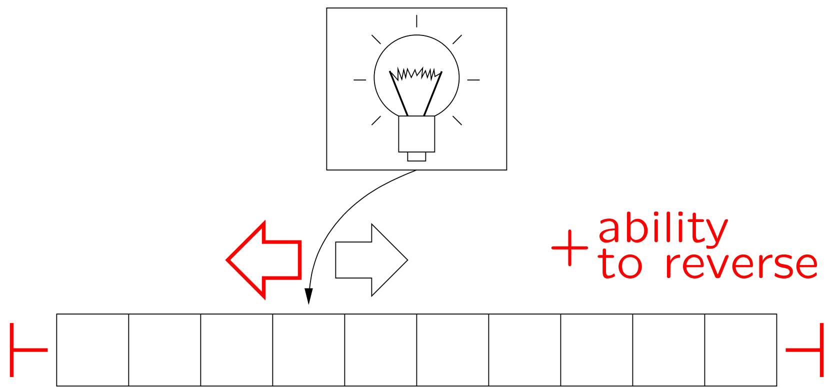
"*the trade-off is exactly* $2^n - 1$"

the main problem

2DFA



2NFA

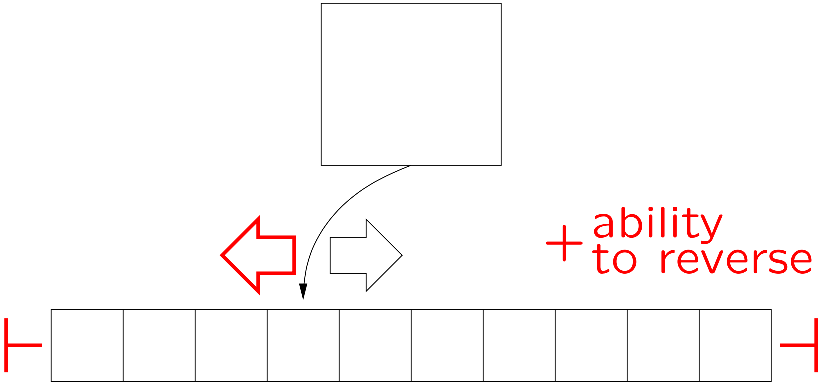


every 1NFA with n states ← can be converted to ← a 1DFA with $\leq 2^n - 1$ states

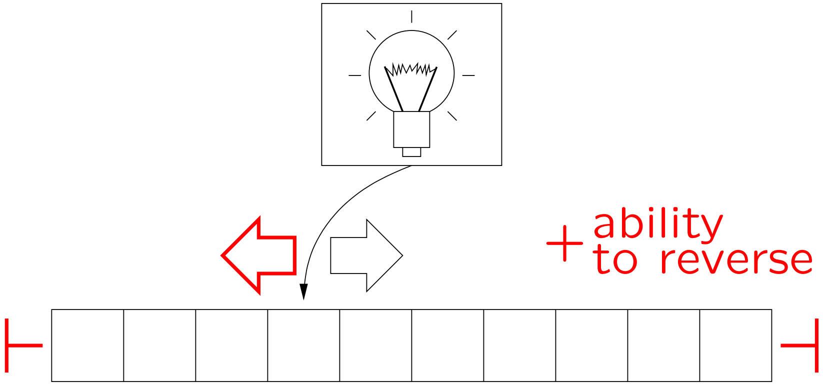
and sometimes all these $2^n - 1$ states are necessary

"the trade-off is exactly $2^n - 1$"

2DFA



2NFA



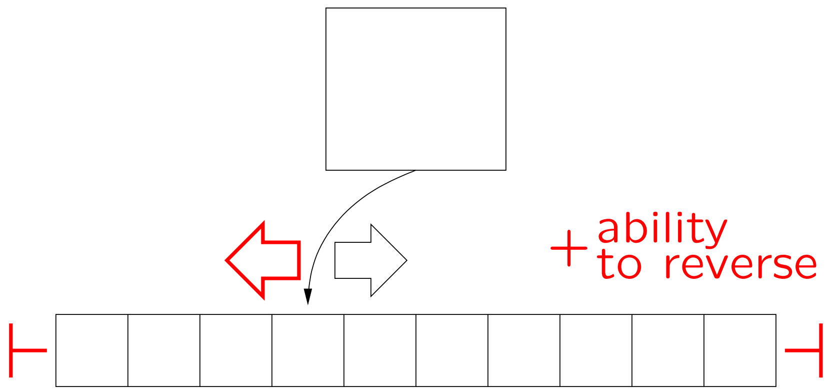
a 2DFA with ≤ ? states ← can be converted to — every 2NFA with n states

and sometimes all these ? states are necessary

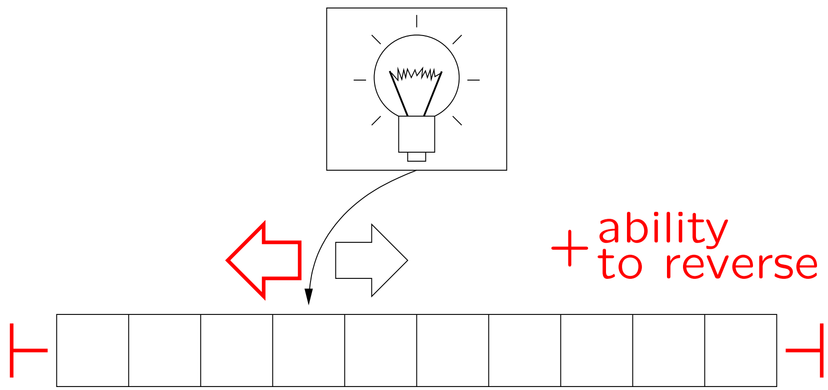
"the trade-off is exactly ? *"*

the main problem

2DFA



2NFA

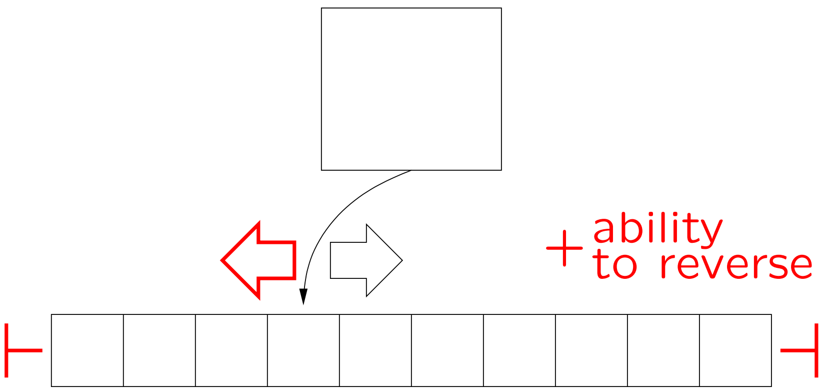


a 2DFA with $\leq$ [?] states ← can be converted to — every 2NFA with n states

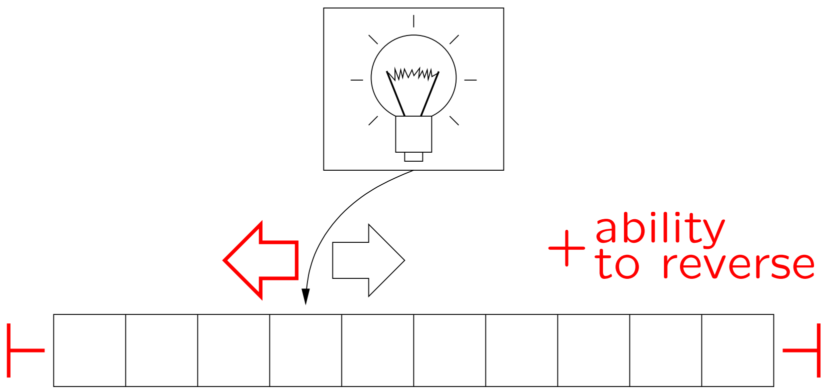
and sometimes all these [?] states are necessary

the trade-off is $\Omega(n^2)$ and $2^{O(n^2)}$

2DFA



2NFA



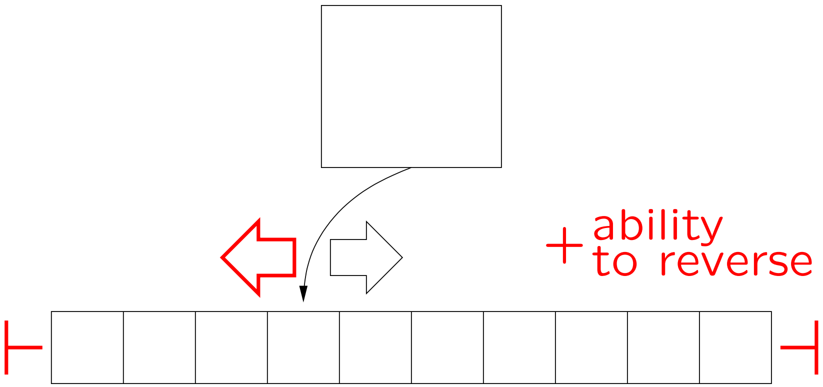
a 2DFA with $\leq$ [?] states ← can be converted to — every 2NFA with n states

and sometimes all these [?] states are necessary

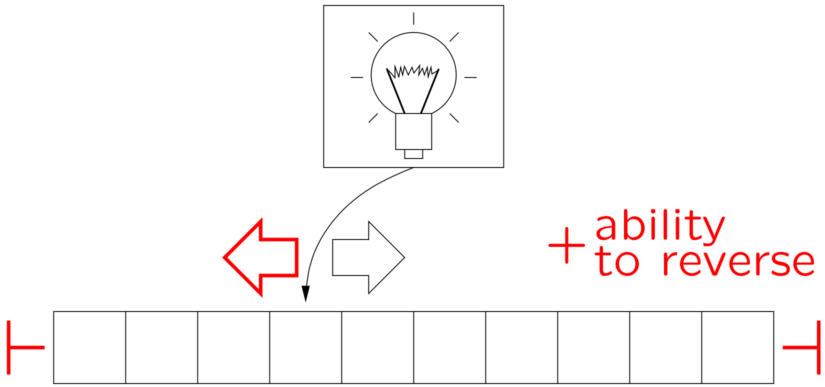
is the trade-off *polynomial*?

the main problem

2DFA



2NFA

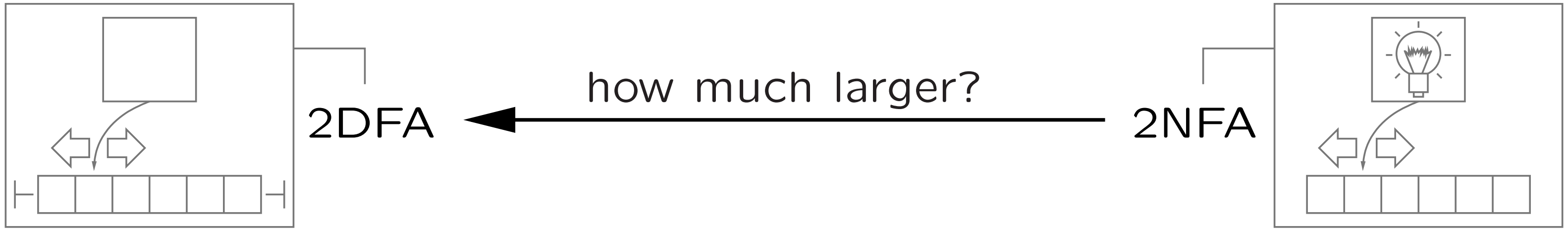


a 2DFA with $\leq$ [?] states ← can be converted to — every 2NFA with n states

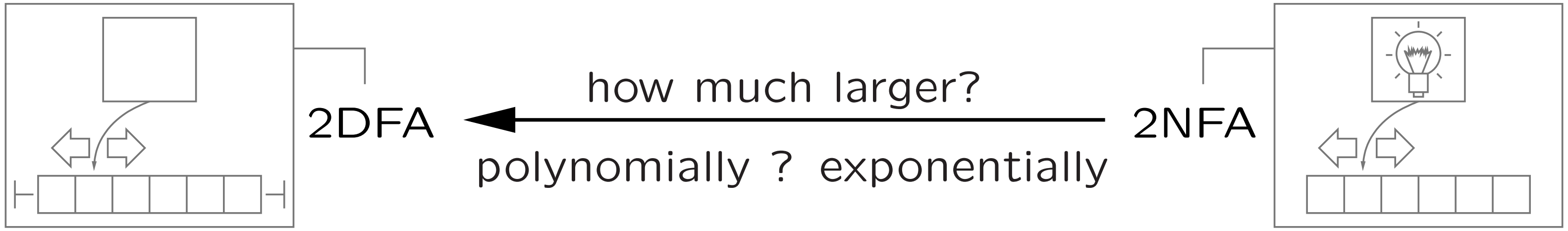
and sometimes all these [?] states are necessary

CONJECTURE: the trade-off from 2NFAs to 2DFAs is exponential

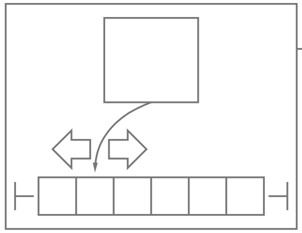
why care


2DFA
how much larger?
2NFA

why care

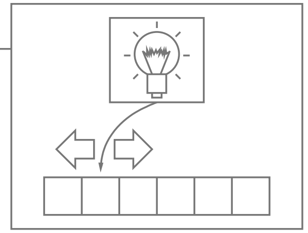


why care



2DFA ← how much larger? — 2NFA

polynomially ? exponentially



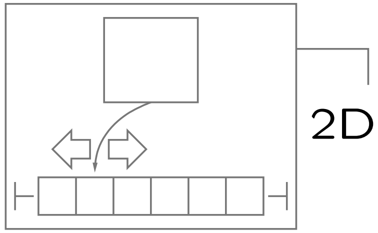
determ Turing machine

DTM ← how much slower? — NTM

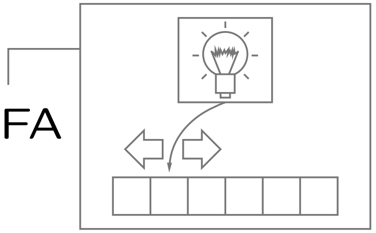
polynomially ? exponentially

nondeterm Turing machine

why care



2DFA ← how much larger? — 2NFA



polynomially ? exponentially

determ Turing machine

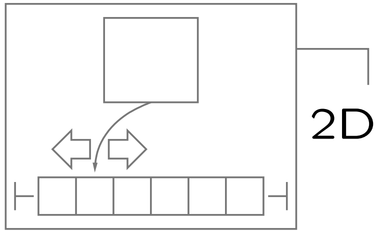
DTM ← how much slower? — NTM

nondeterm Turing machine

polynomially ? exponentially

P=NP P≠NP

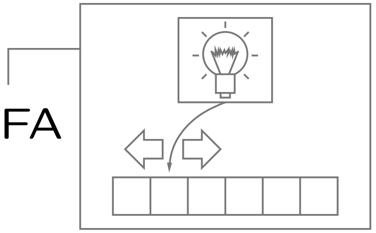
why care



2DFA ← how much larger? — 2NFA

polynomially ? exponentially

2D=2N   2D≠2N



determ Turing machine

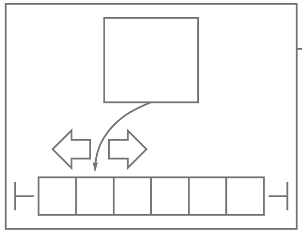
DTM ← how much slower? — NTM

polynomially ? exponentially

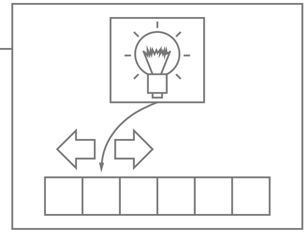
P=NP   P≠NP

nondeterm Turing machine

why care



2DFA ← how much larger? ← 2NFA



polynomially ? exponentially

2D=2N   2D≠2N

determ Turing machine

DTM ← how much slower? ← NTM

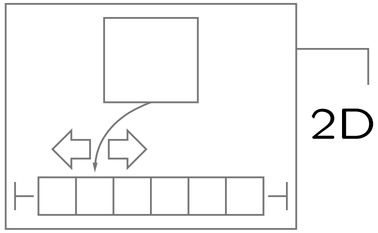
nondeterm Turing machine

polynomially ? exponentially

P=NP   P≠NP

is *satisfiability* in P?

why care



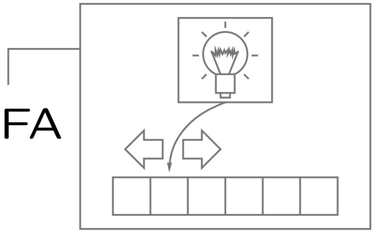
2DFA ← 2NFA

how much larger?

polynomially ? exponentially

2D=2N 2D≠2N

is *two-way liveness* in 2D?



determ Turing machine

DTM ← NTM

how much slower?

polynomially ? exponentially

P=NP P≠NP

is *satisfiability* in P?

nondeterm Turing machine

[Seiferas73]	question posed, hard problems small *single-pass* 2DFAs cannot solve one-way liveness
[BermanLingas77]	if 2D≠2N on *short* inputs, then L≠NL
[SakodaSipser78]	complexity classes, reductions, complete problems
[Sipser79]	small *sweeping* 2DFAs cannot solve one-way liveness
[Berman80] [Micali81]	full 2DFAs can be much smaller than sweeping ones
[Kannan83]	under *positional simulation*, the trade-off is $2^{\Omega(\lg^k n)}$
[Chrobak86]	the $\Omega(n^2)$ lower bound (even from unary 1NFAs)
[Birget92]	positional simulation of 2NFAs by 2DFAs is always possible
[Leung01]	separation of [S79] holds even on *binary* alphabets
[GeffertMereghettiPighizzini03]	on *unary* inputs, the trade-off is $2^{O(\lg^2 n)}$
[HromkovicSchnitger03]	small *oblivious* 2DFAs cannot solve one-way liveness
[K05]	deterministic *moles* cannot solve one-way liveness
[GeffertMereghettiPighizzini05]	small *unary* 2NFAs are closed under complement

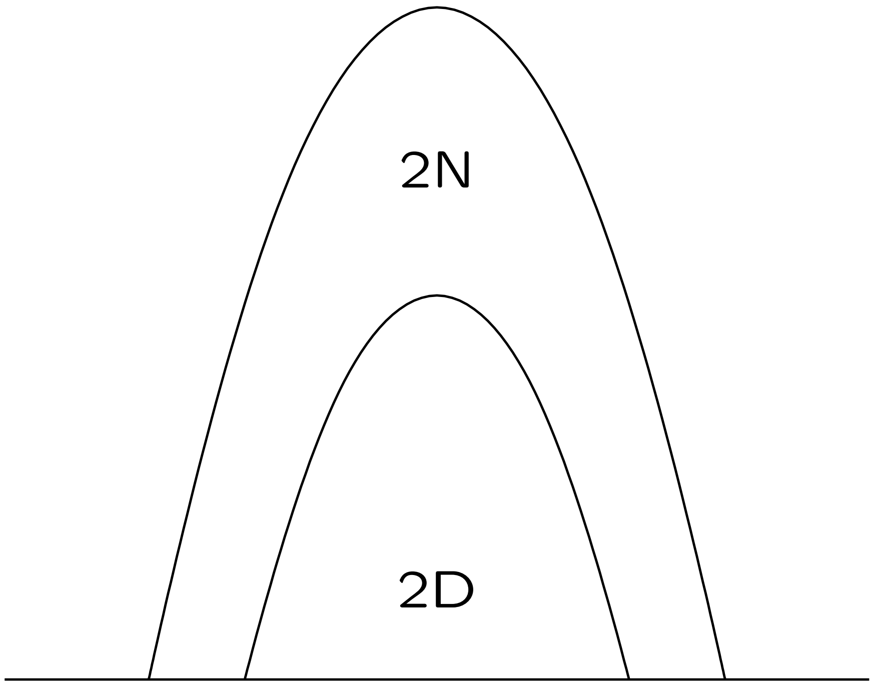
why care

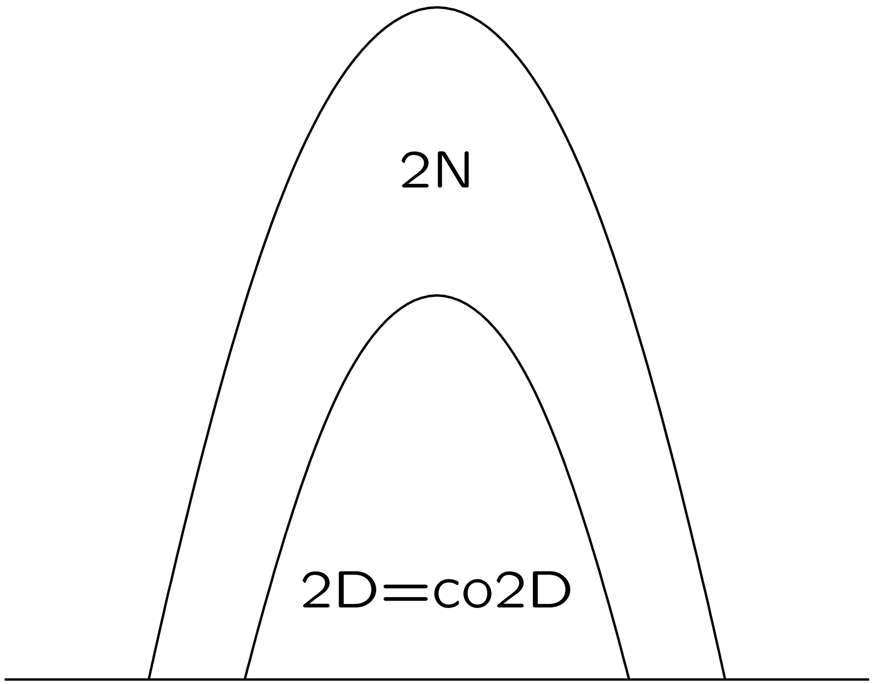
[Seiferas73]	question posed, hard problems small *single-pass* 2DFAs cannot solve one-way liveness
[BermanLingas77]	if 2D≠2N on *short* inputs, then L≠NL
[SakodaSipser78]	complexity classes, reductions, complete problems
[Sipser79]	small *sweeping* 2DFAs cannot solve one-way liveness
[Berman80] [Micali81]	full 2DFAs can be much smaller than sweeping ones
[Kannan83]	under *positional simulation*, the trade-off is $2^{\Omega(\lg^k n)}$
[Chrobak86]	the $\Omega(n^2)$ lower bound (even from unary 1NFAs)
[Birget92]	positional simulation of 2NFAs by 2DFAs is always possible
[Leung01]	separation of [S79] holds even on *binary* alphabets
[GeffertMereghettiPighizzini03]	on *unary* inputs, the trade-off is $2^{O(\lg^2 n)}$
[HromkovicSchnitger03]	small *oblivious* 2DFAs cannot solve one-way liveness
[K05]	deterministic *moles* cannot solve one-way liveness
[GeffertMereghettiPighizzini05]	small *unary* 2NFAs are closed under complement

why care

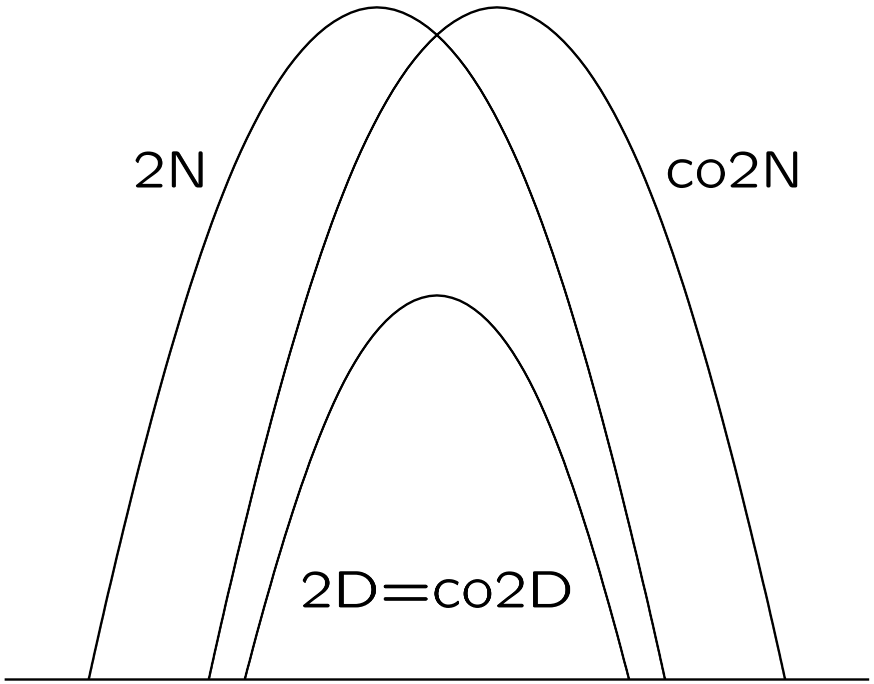
[Seiferas73]	question posed, hard problems small *single-pass* 2DFAs cannot solve one-way liveness
[BermanLingas77]	if 2D≠2N on *short* inputs, then L≠NL
[SakodaSipser78]	complexity classes, reductions, complete problems
[Sipser79]	small *sweeping* 2DFAs cannot solve one-way liveness
[Berman80] [Micali81]	full 2DFAs can be much smaller than sweeping ones
[Kannan83]	under *positional simulation*, the trade-off is $2^{\Omega(\lg^k n)}$
[Chrobak86]	the $\Omega(n^2)$ lower bound (even from unary 1NFAs)
[Birget92]	positional simulation of 2NFAs by 2DFAs is always possible
[Leung01]	separation of [S79] holds even on *binary* alphabets
[GeffertMereghettiPighizzini03]	on *unary* inputs, the trade-off is $2^{O(\lg^2 n)}$
[HromkovicSchnitger03]	small *oblivious* 2DFAs cannot solve one-way liveness
[K05]	deterministic *moles* cannot solve one-way liveness
[GeffertMereghettiPighizzini05]	small *unary* 2NFAs are closed under complement

about sweeping 2NFAs. . .

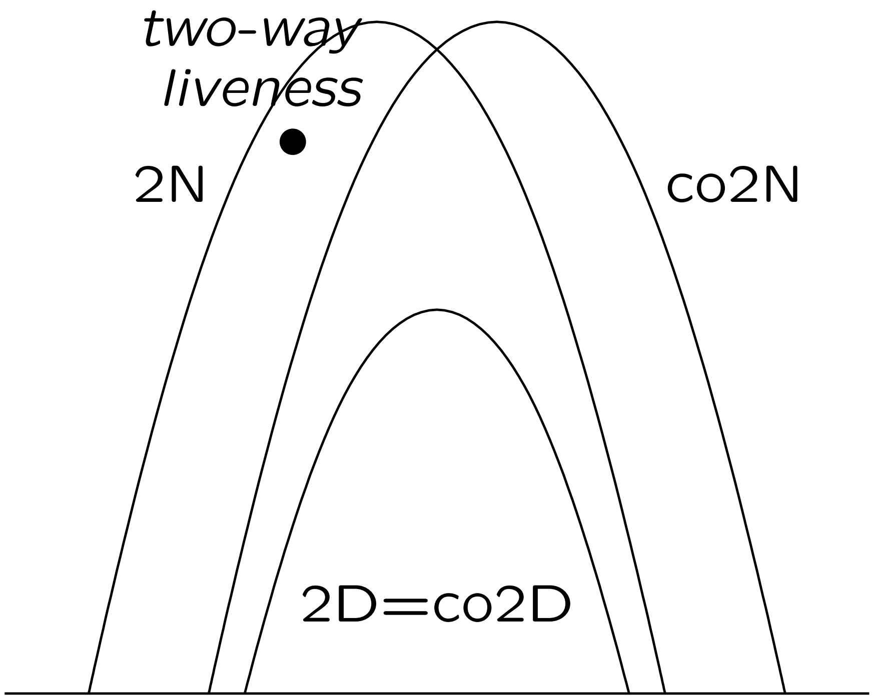

2N
2D



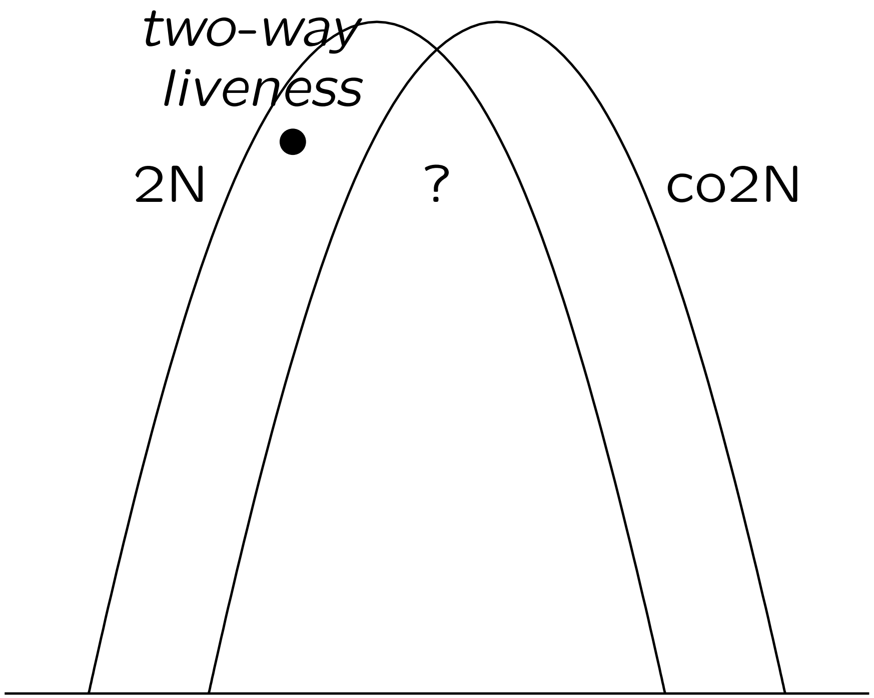
[S80,GMP05]

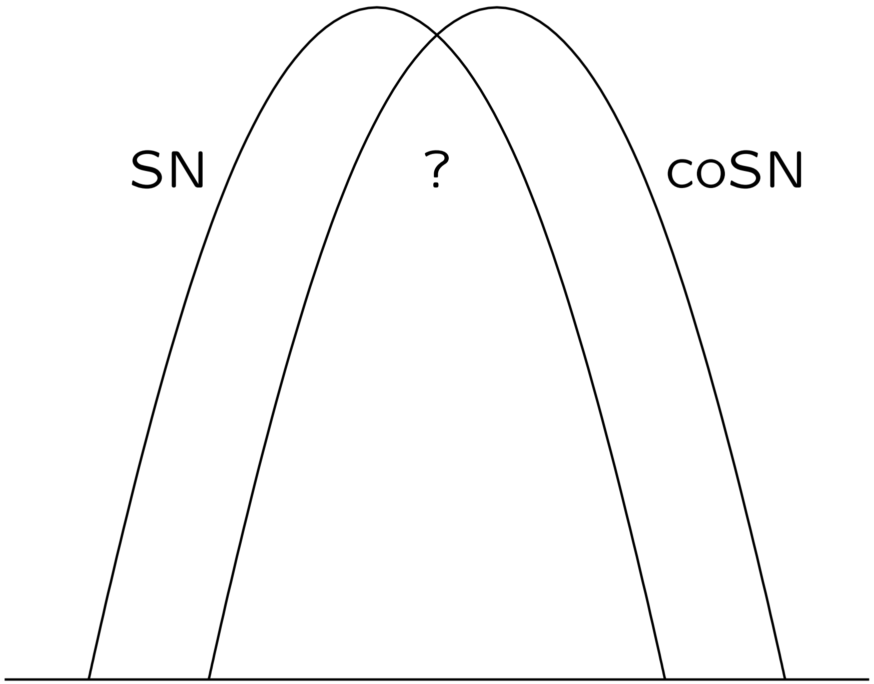


[S80,GMP05]



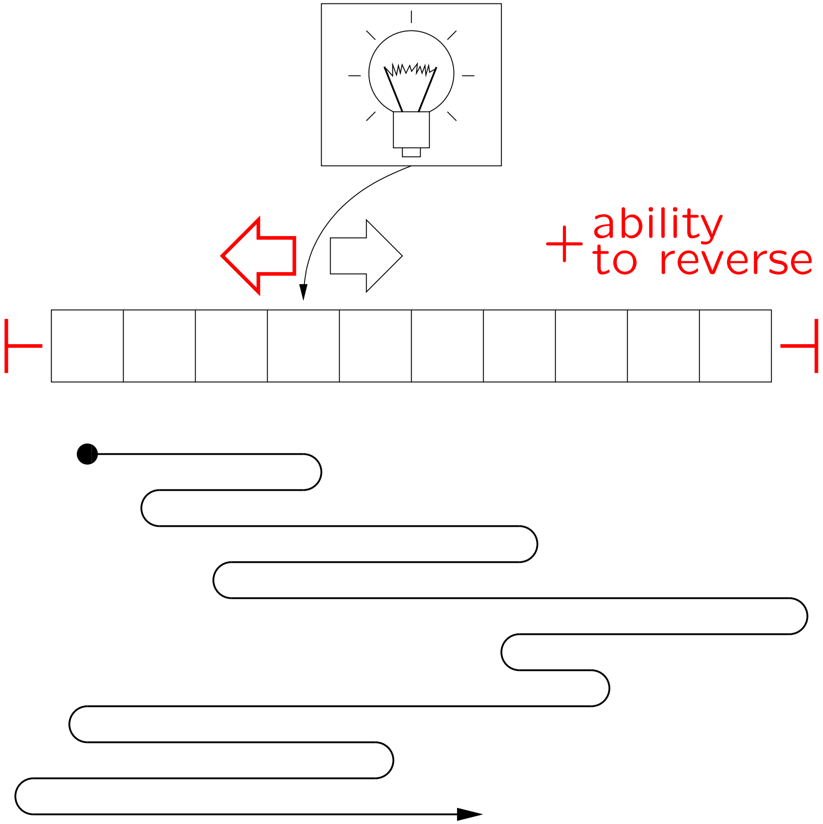
[S80,GMP05]


two-way
liveness
2N
?
co2N

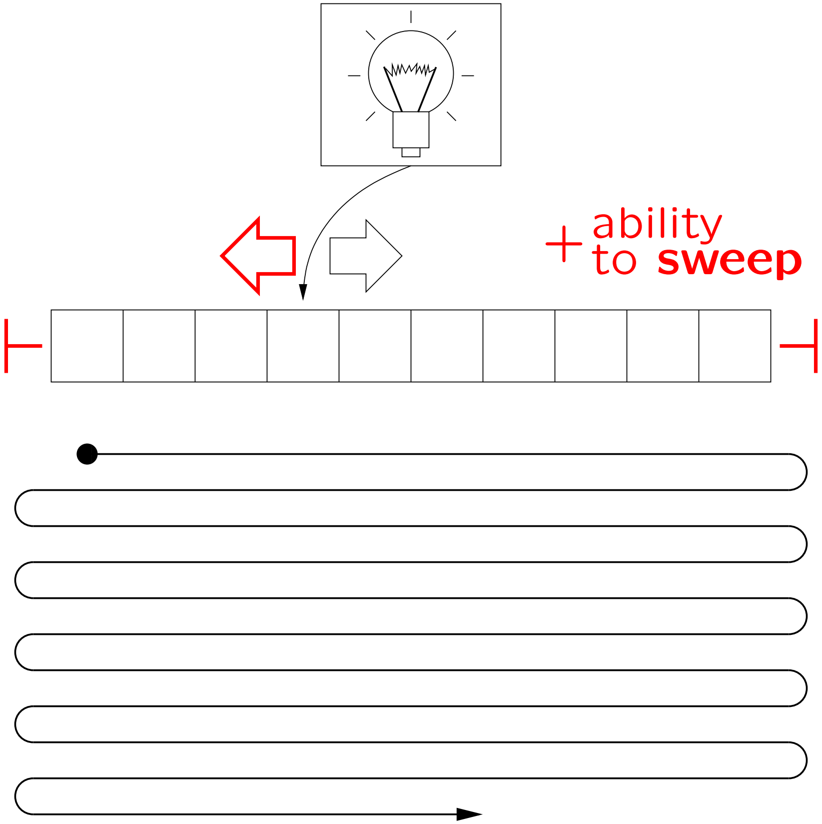


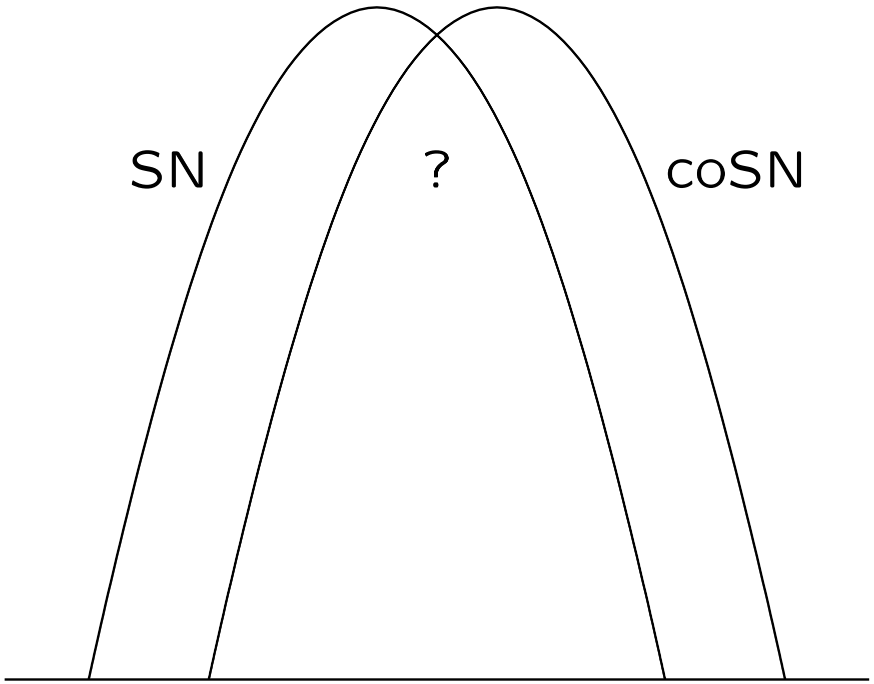
What about just *sweeping* automata?

2NFA

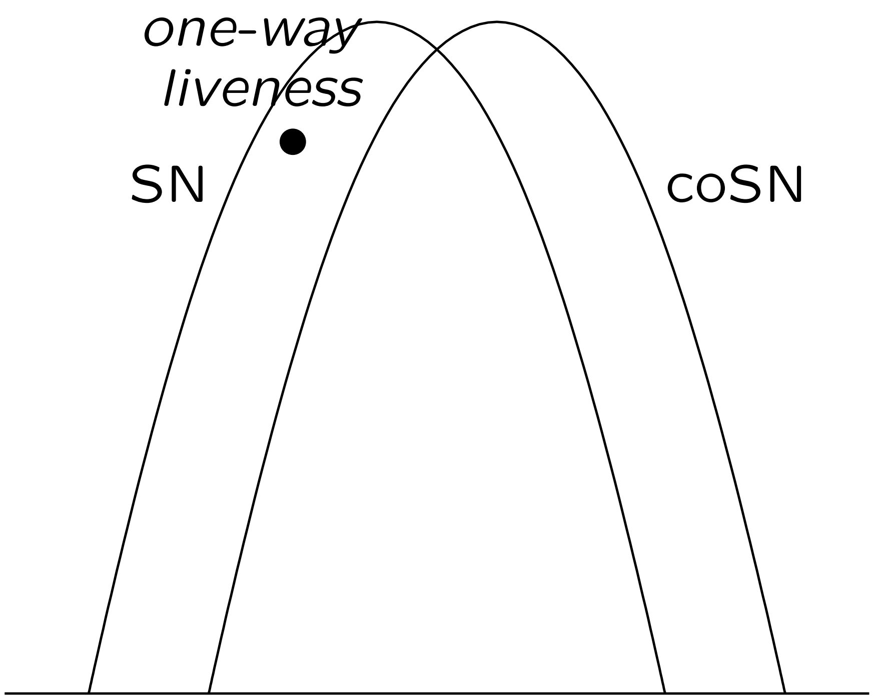


SNFA (sweeping 2NFA)



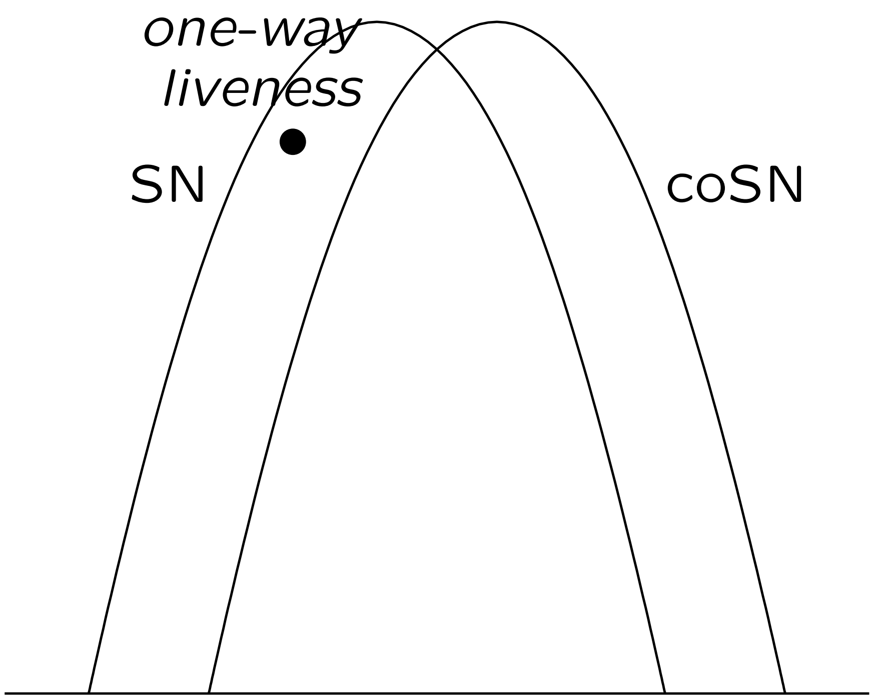


What about just *sweeping* automata?



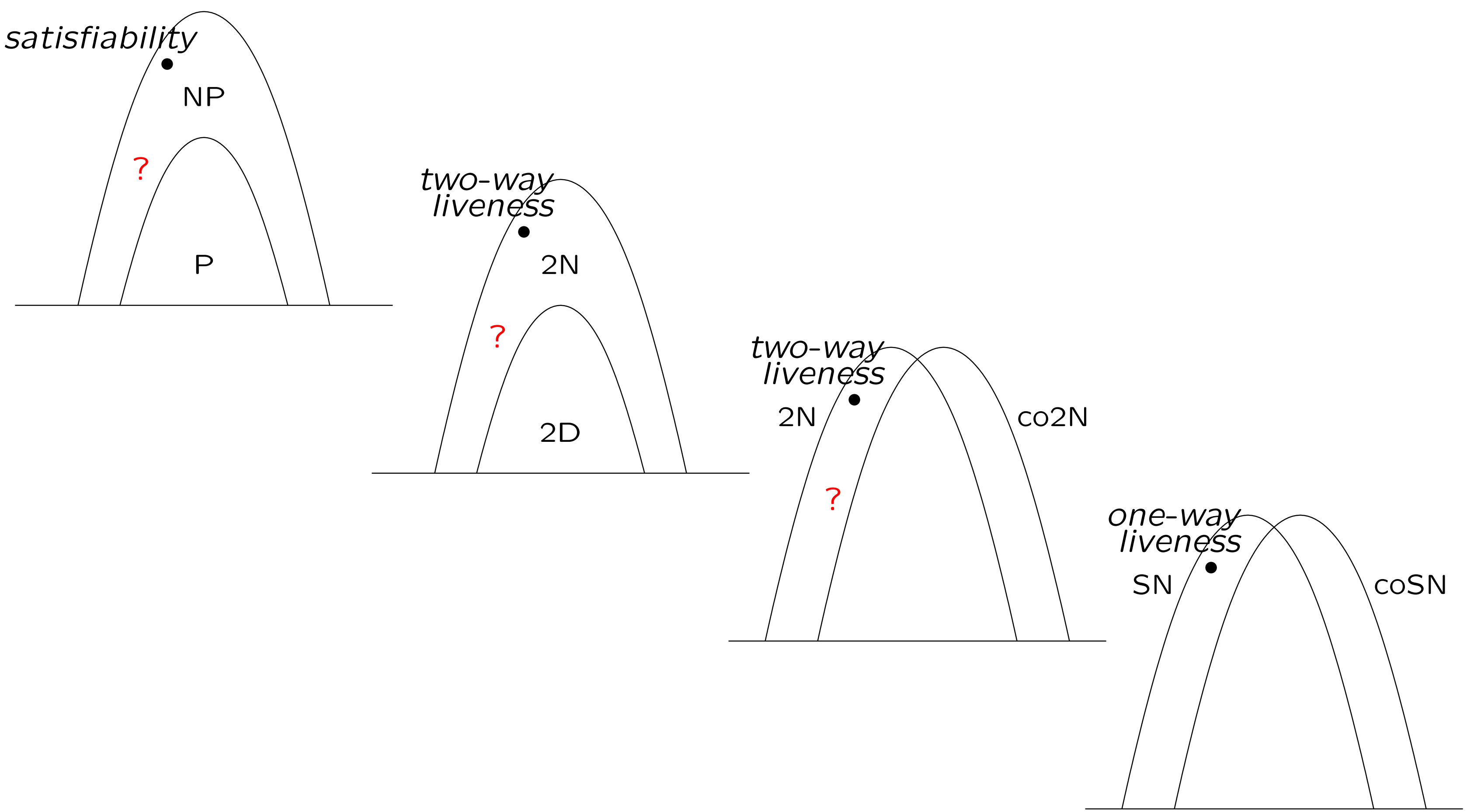
What about just *sweeping* automata?

THEOREM. In the *sweeping* case: SN$\neq$coSN.



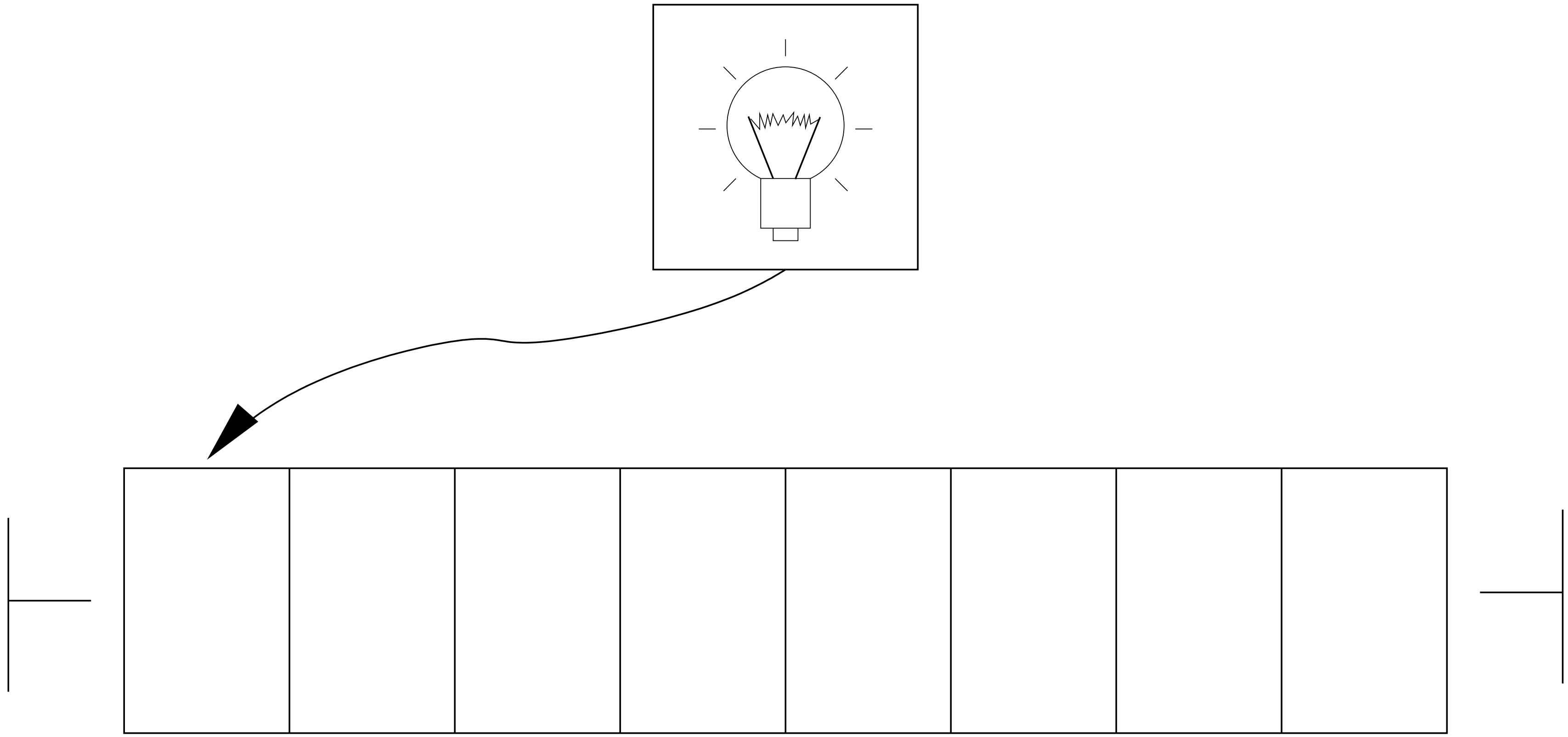
What about just *sweeping* automata?

no small *sweeping* 2NFA can solve the *complement* of one-way liveness

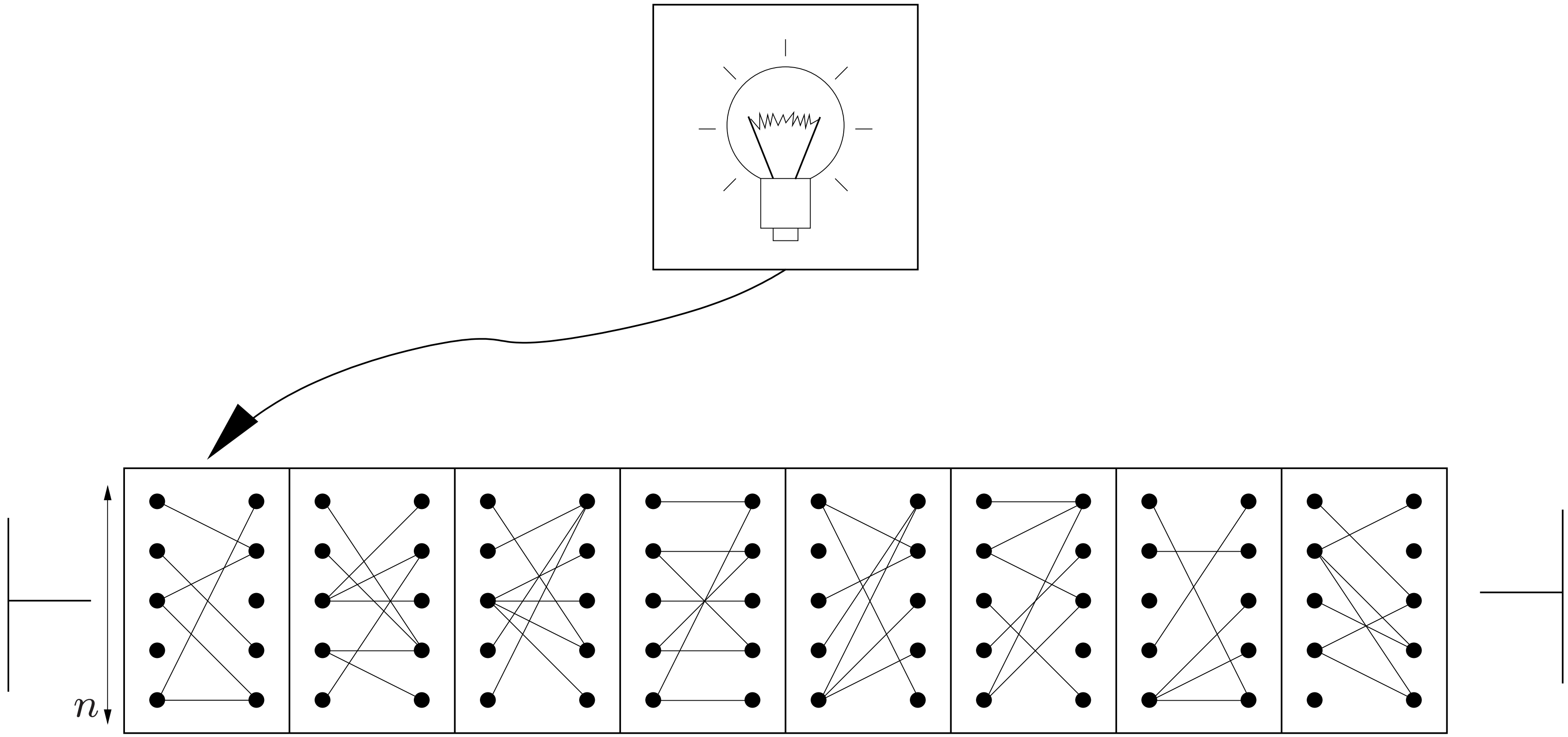


no small *sweeping* 2NFA can solve the *complement* of one-way liveness

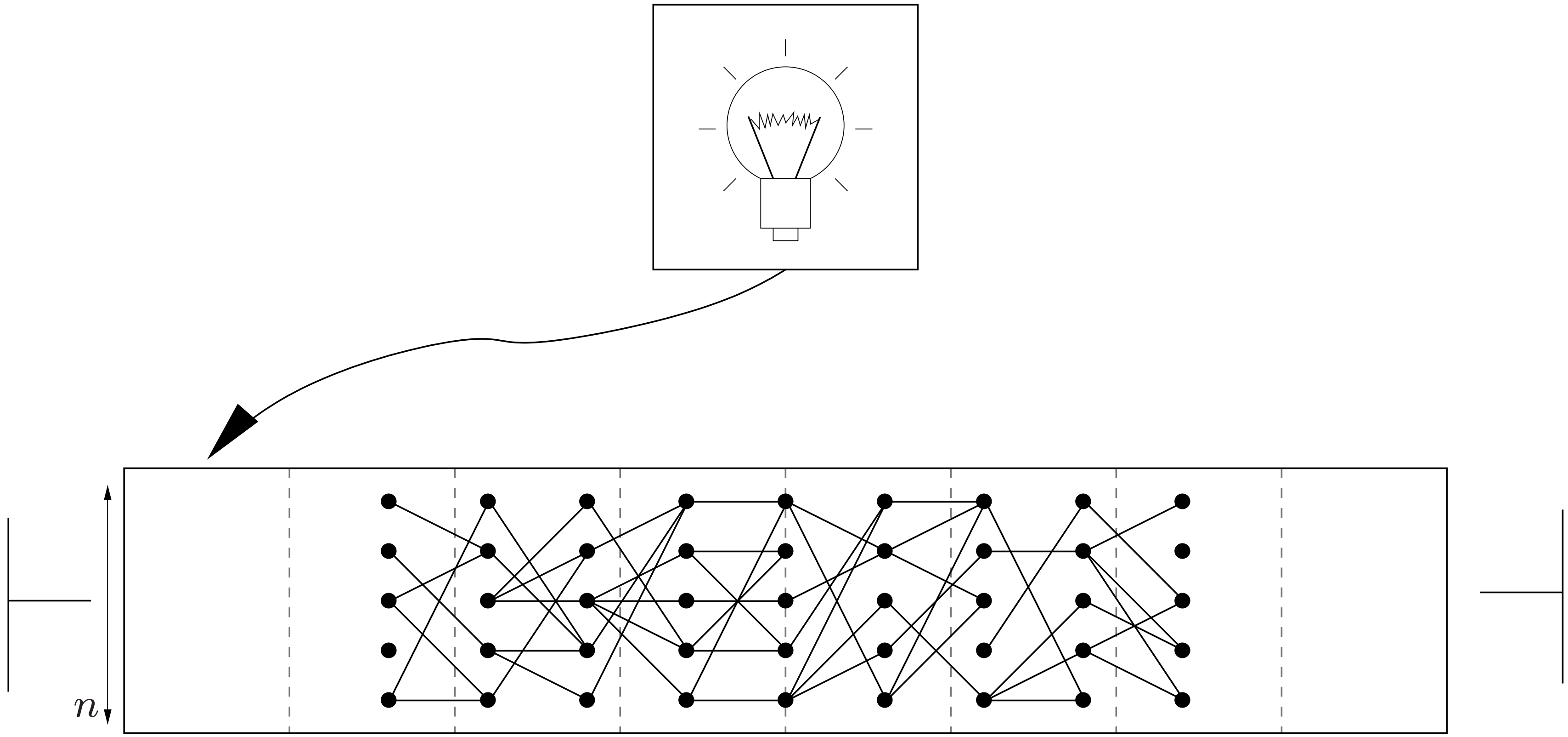
one-way liveness



one-way liveness

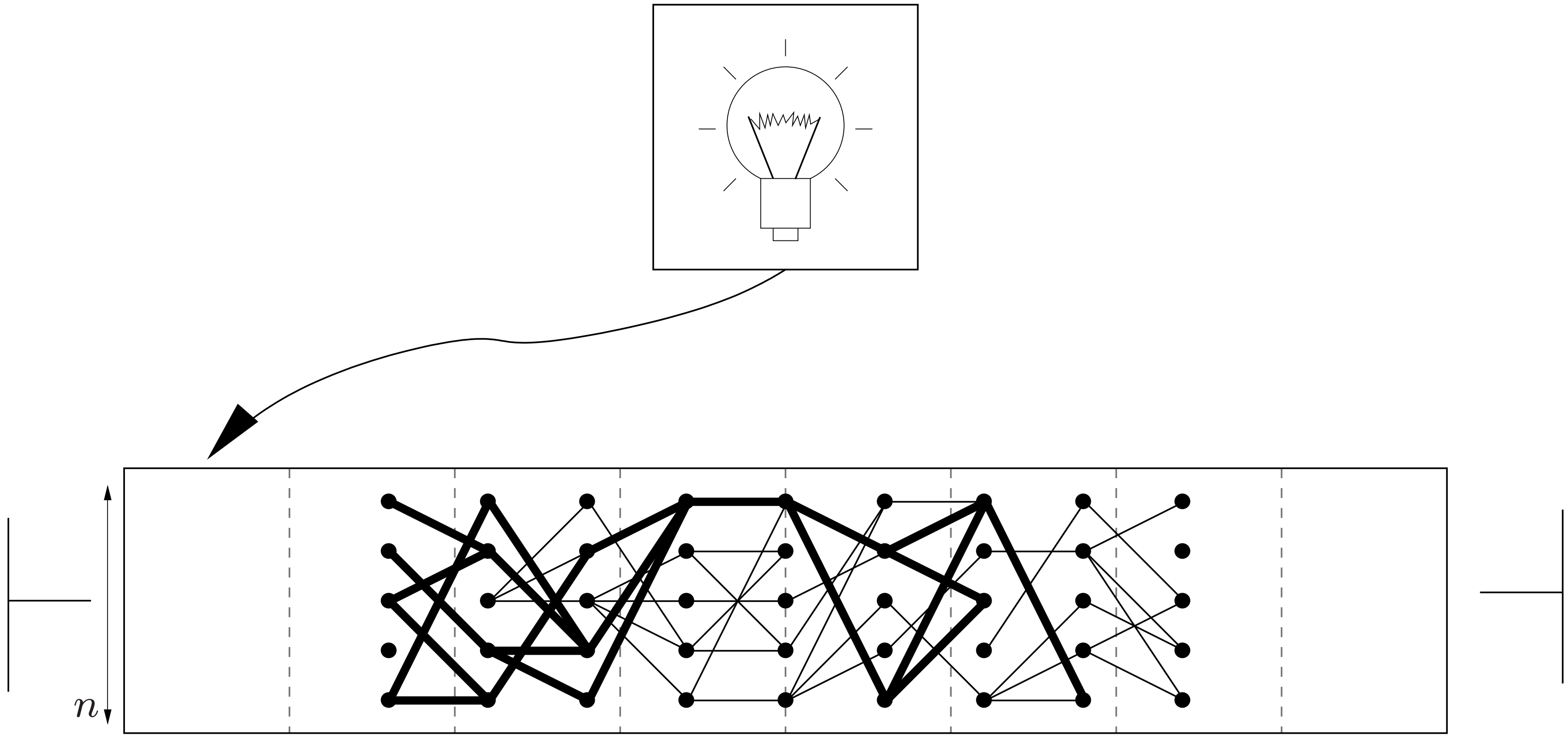


(input alphabet has 2^{n^2} symbols)



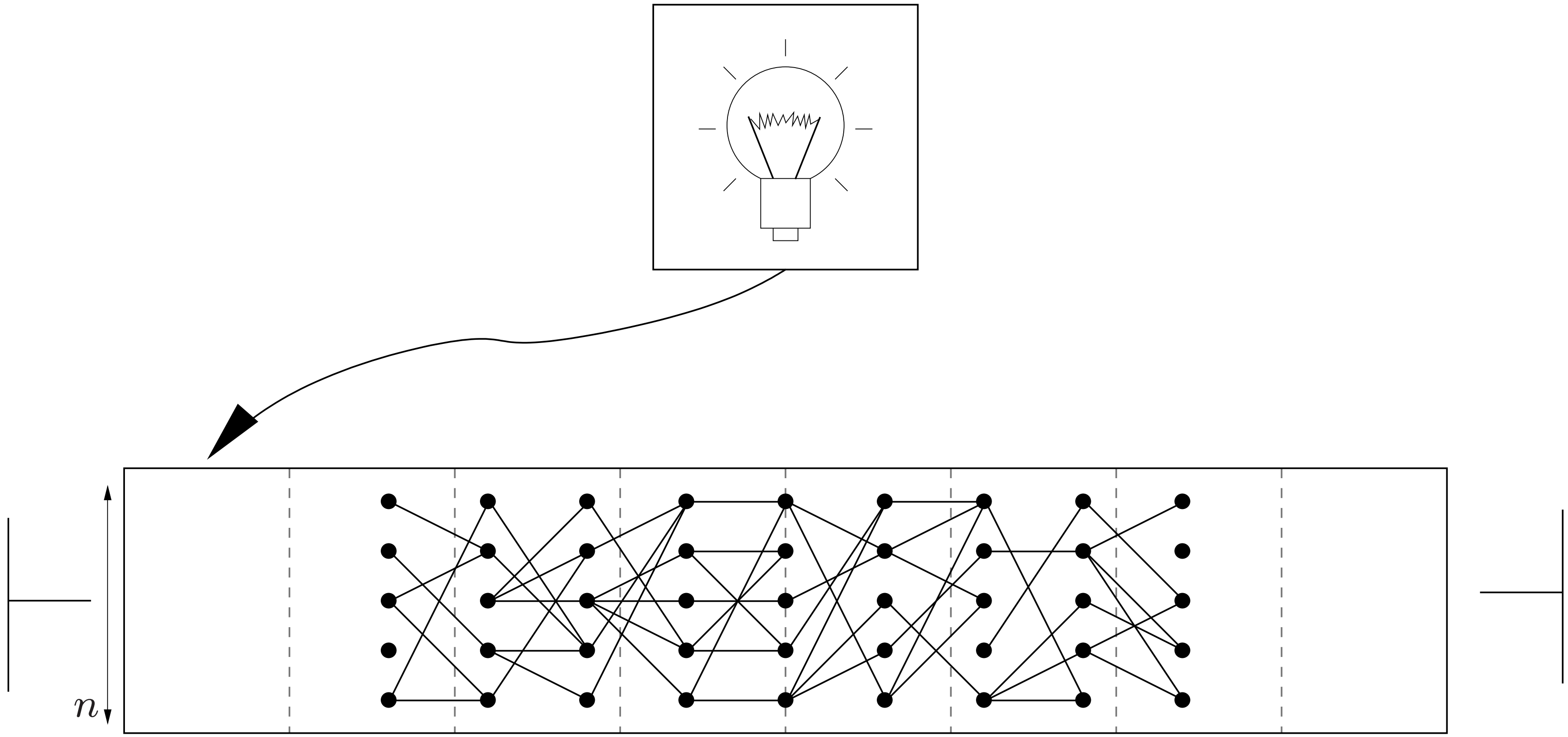
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



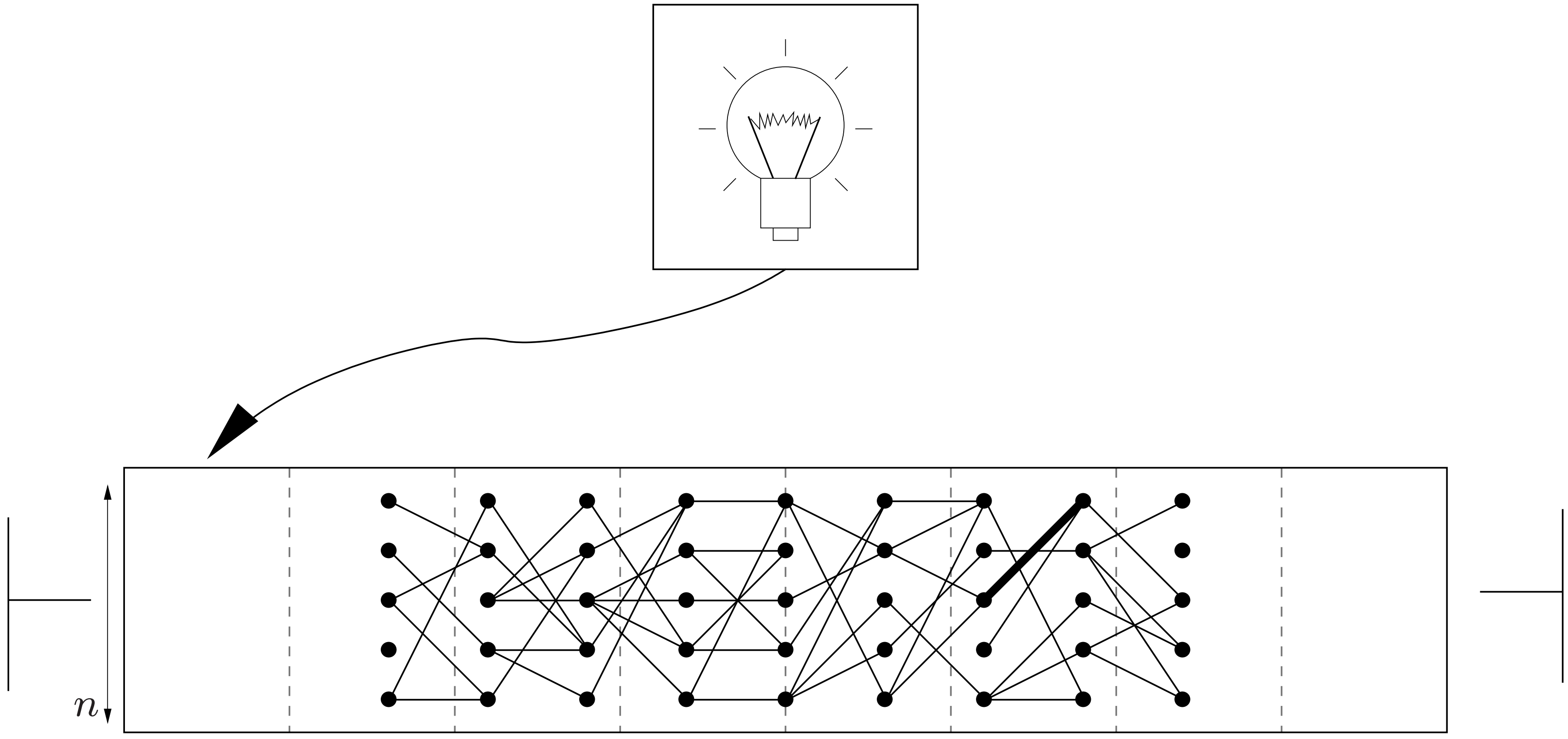
is there a *live* path? no.

(input alphabet has 2^{n^2} symbols)



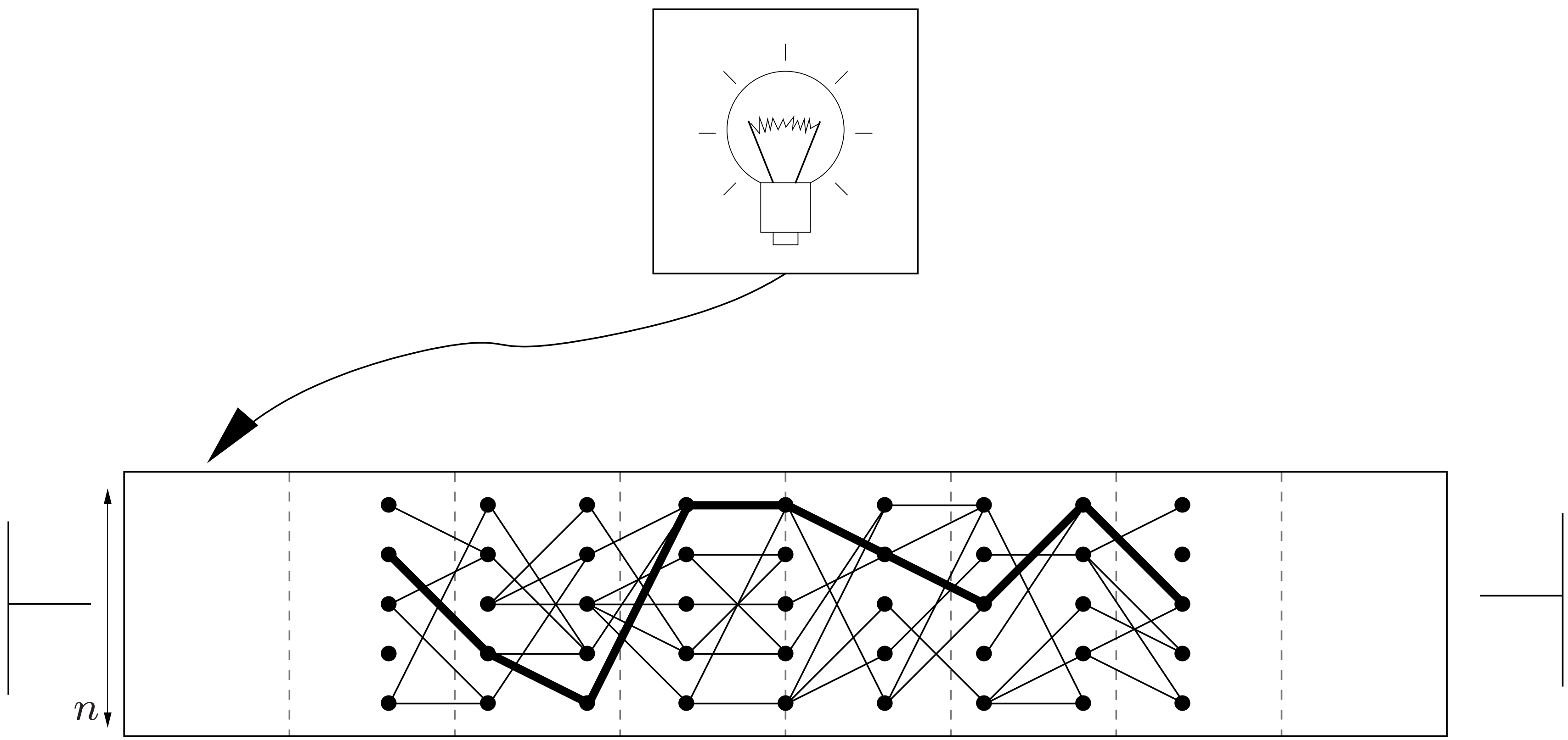
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



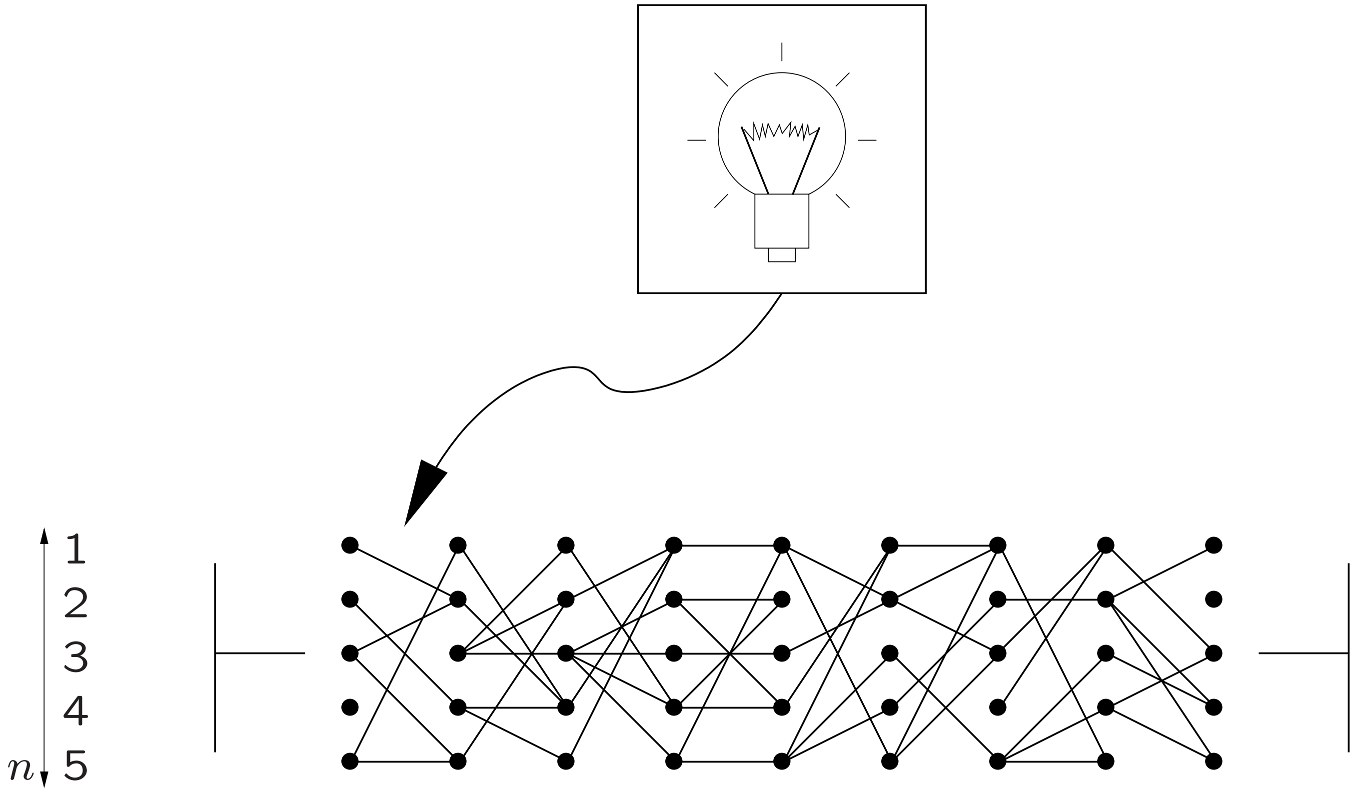
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



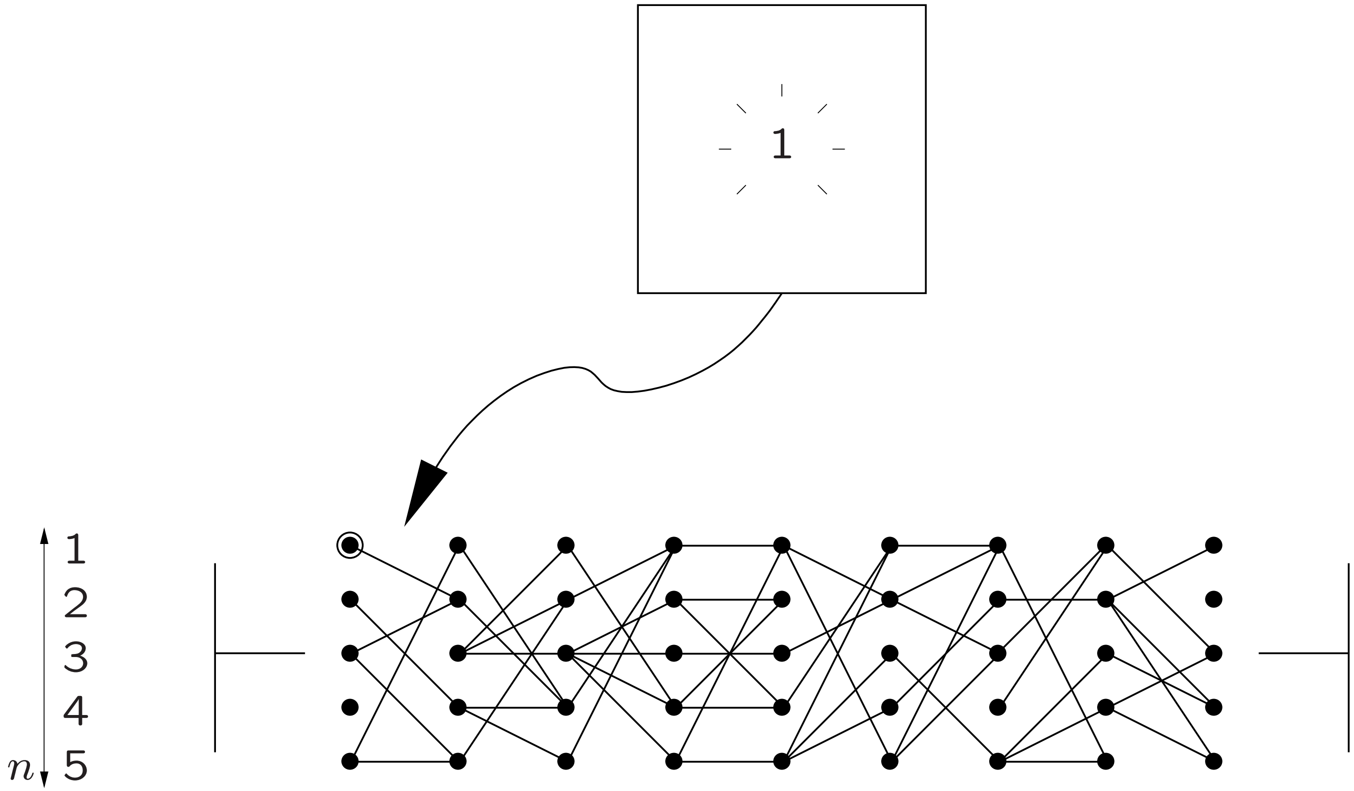
is there a *live* path? yes.

(input alphabet has 2^{n^2} symbols)



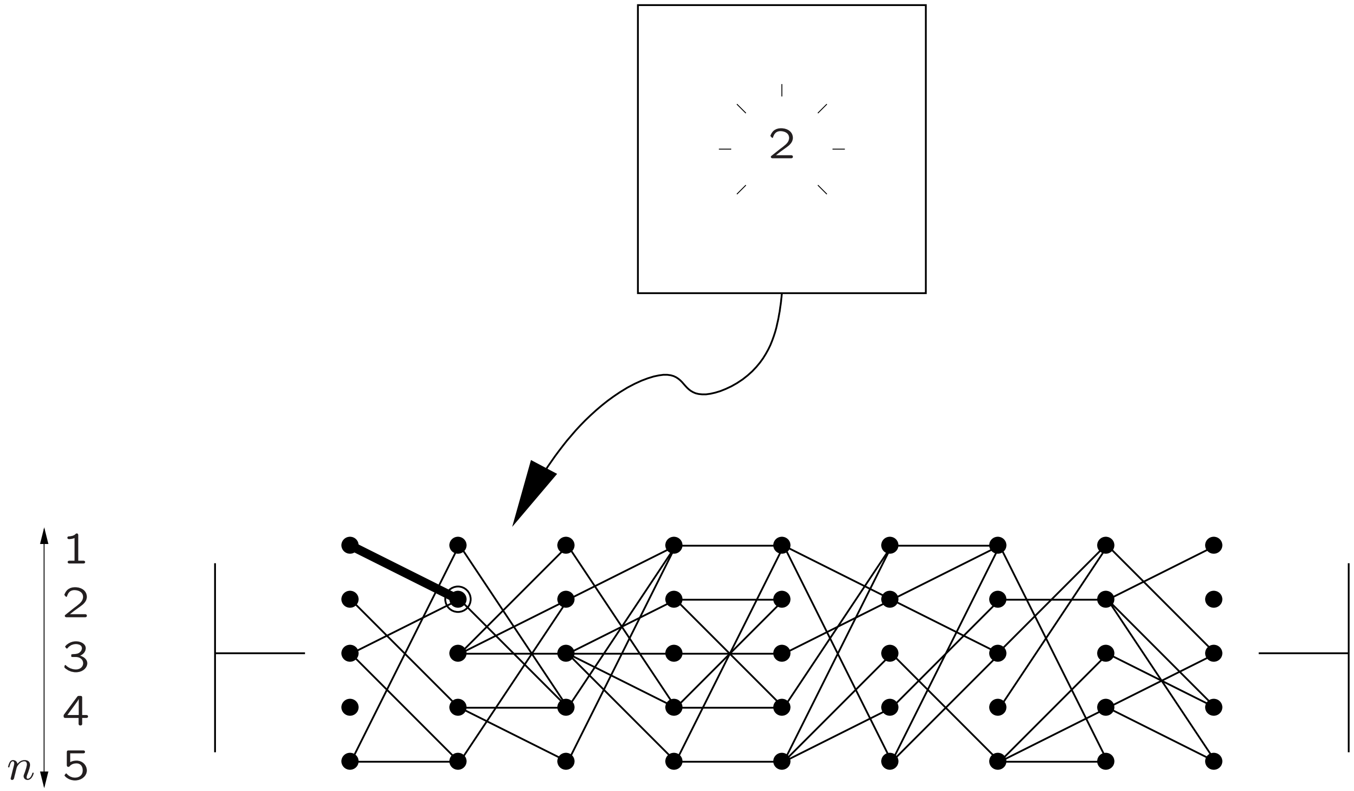
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



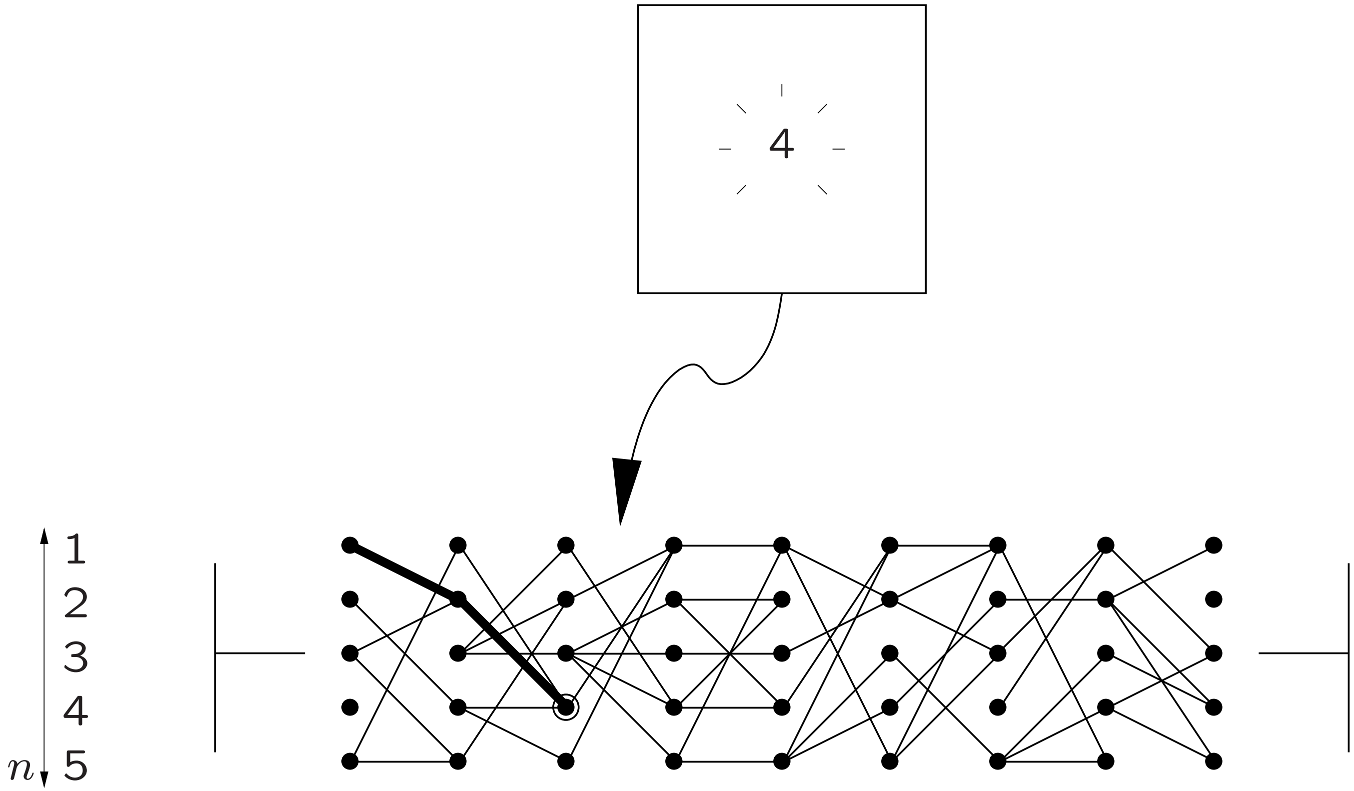
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



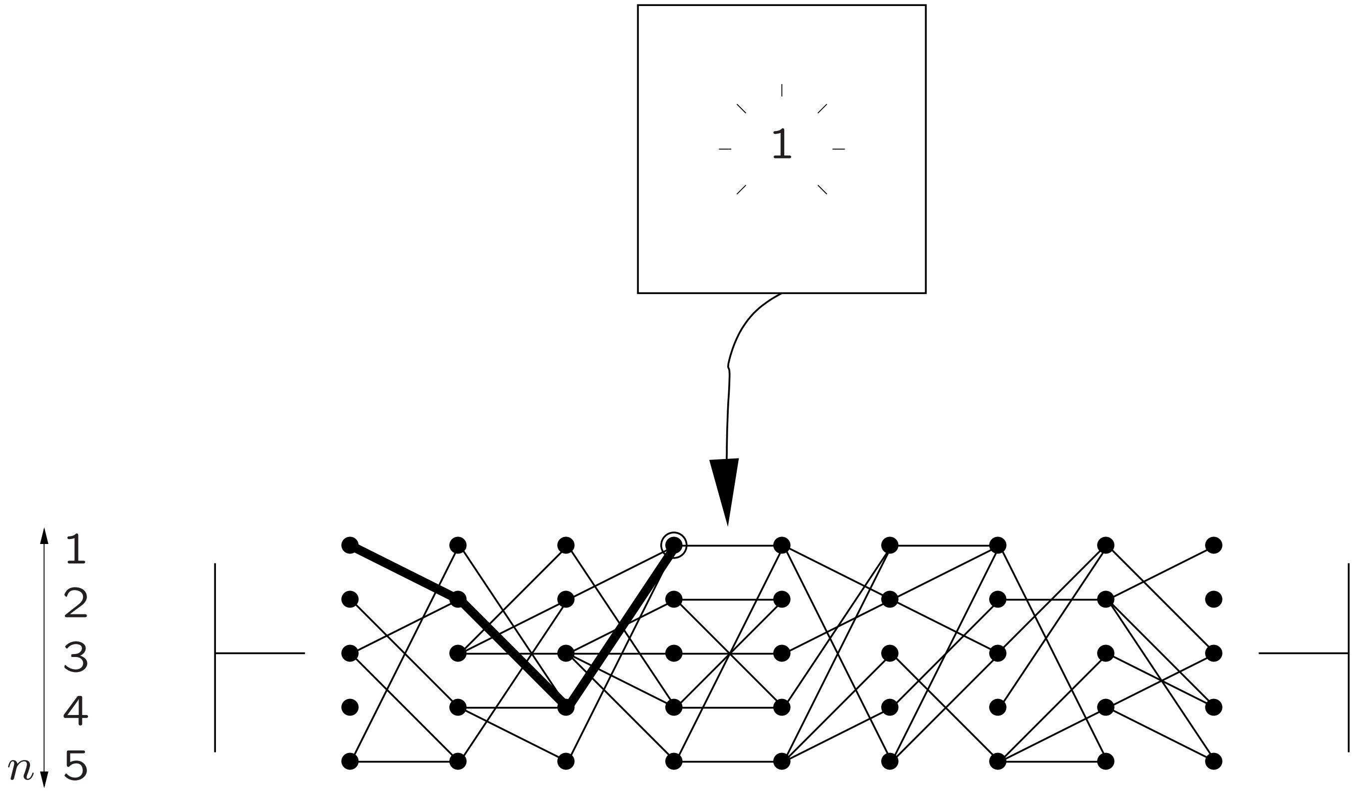
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



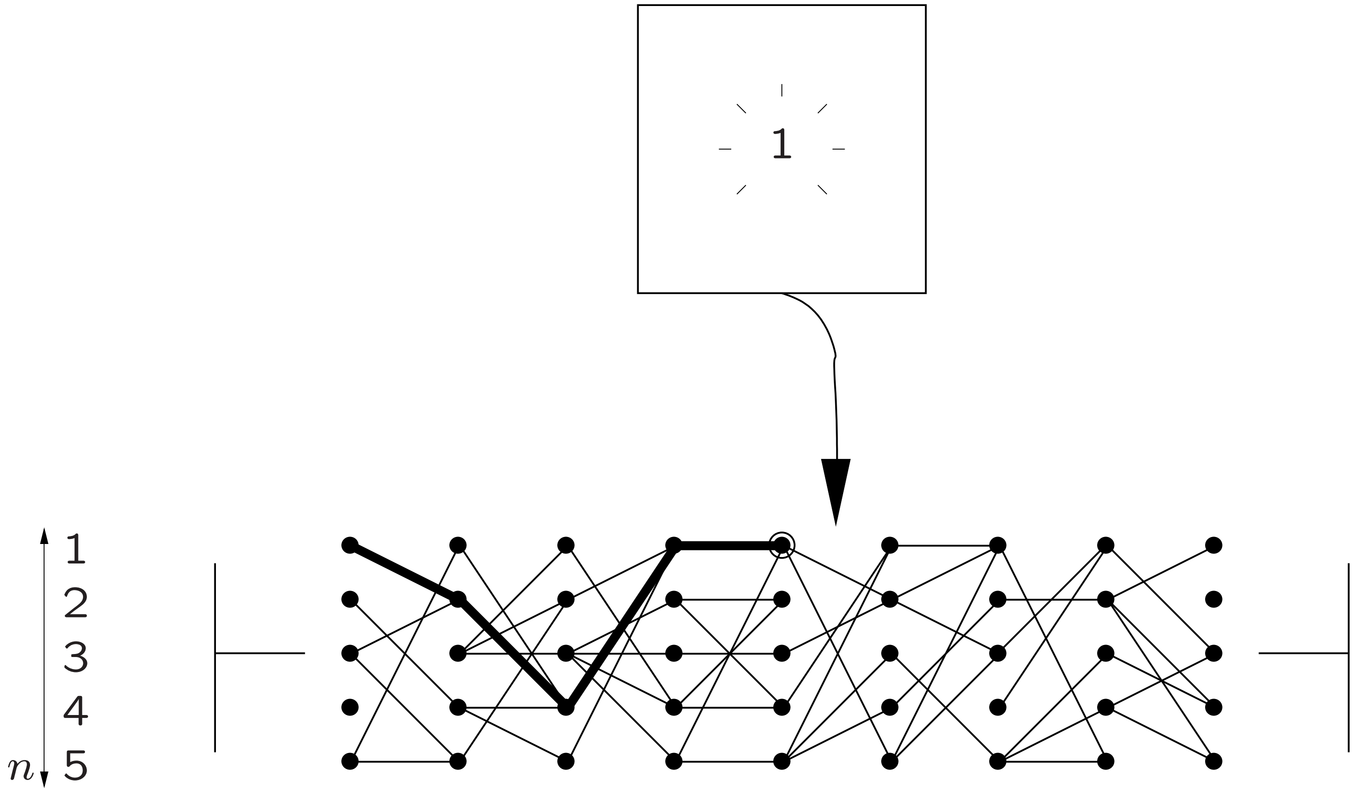
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



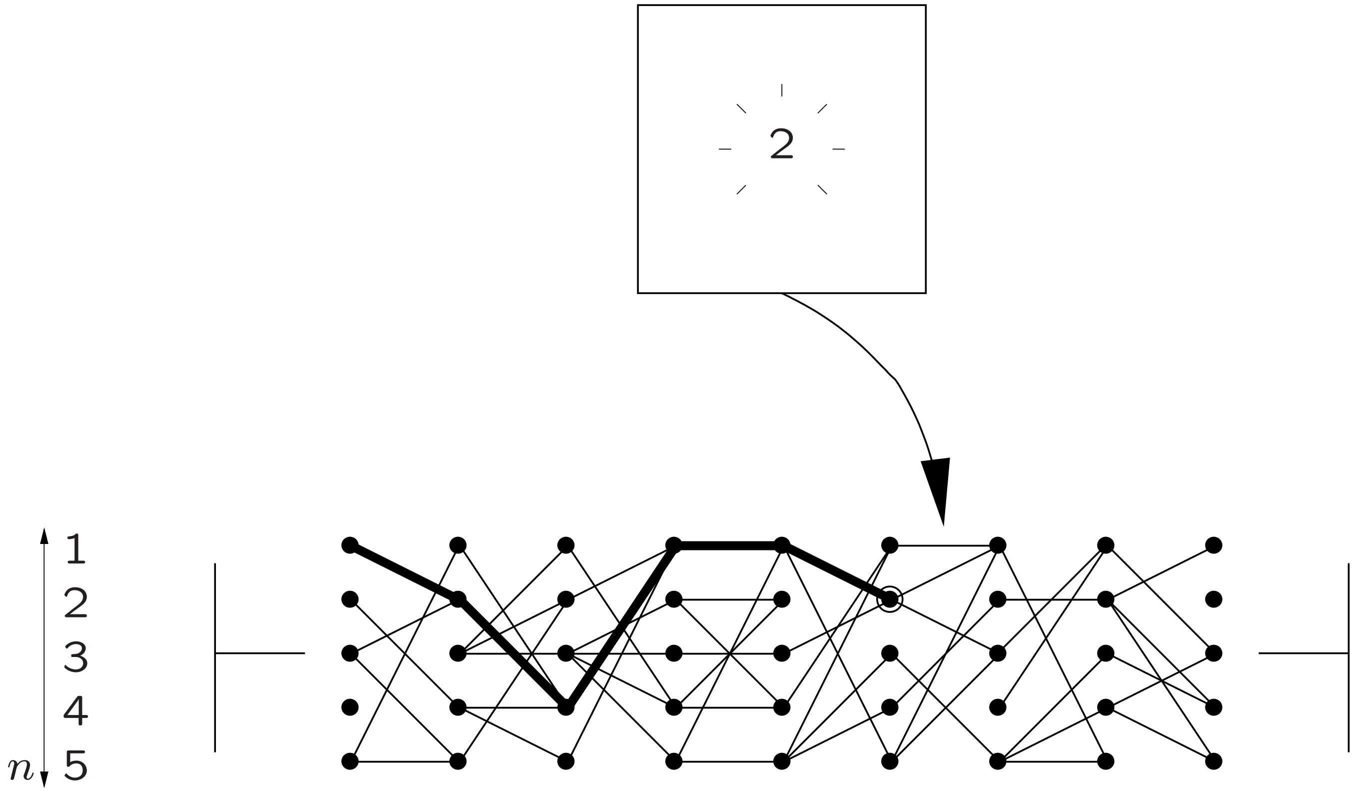
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



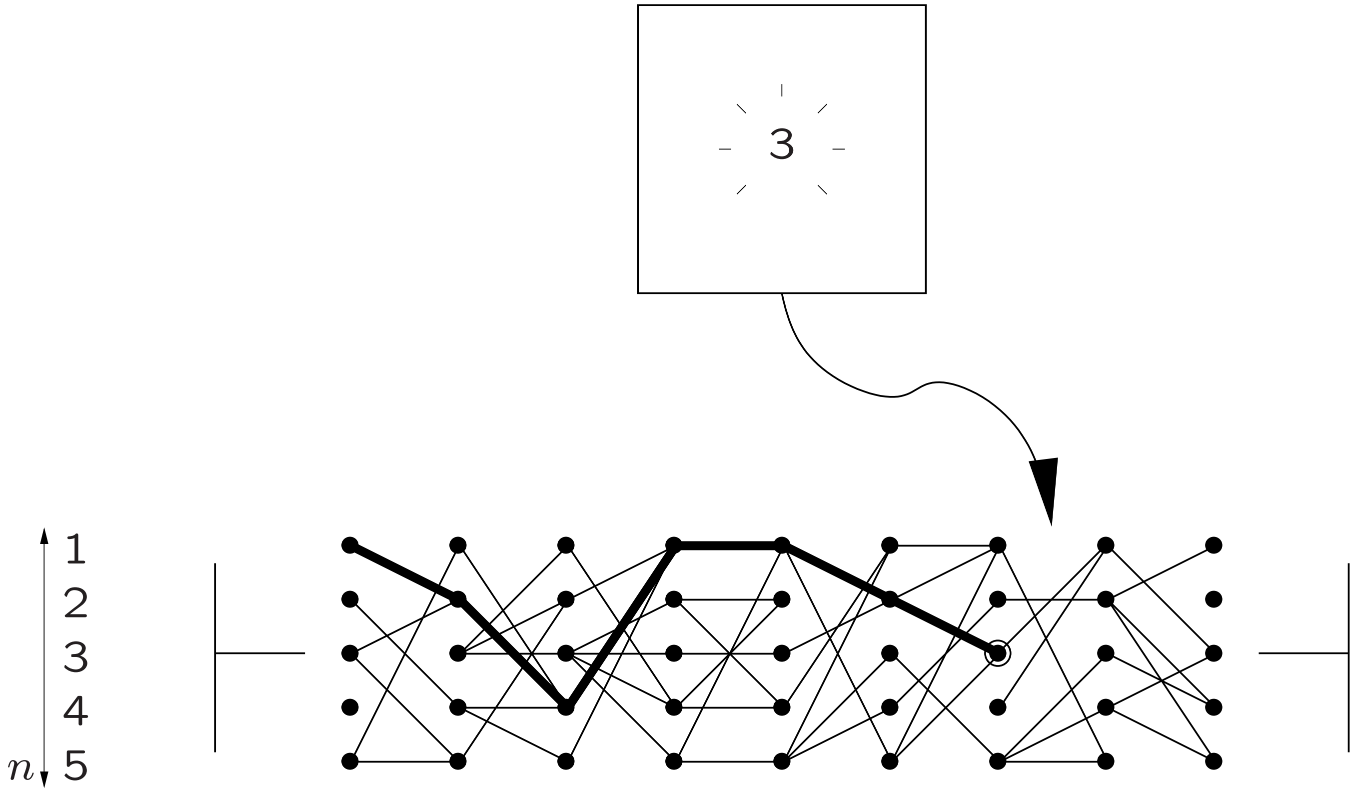
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



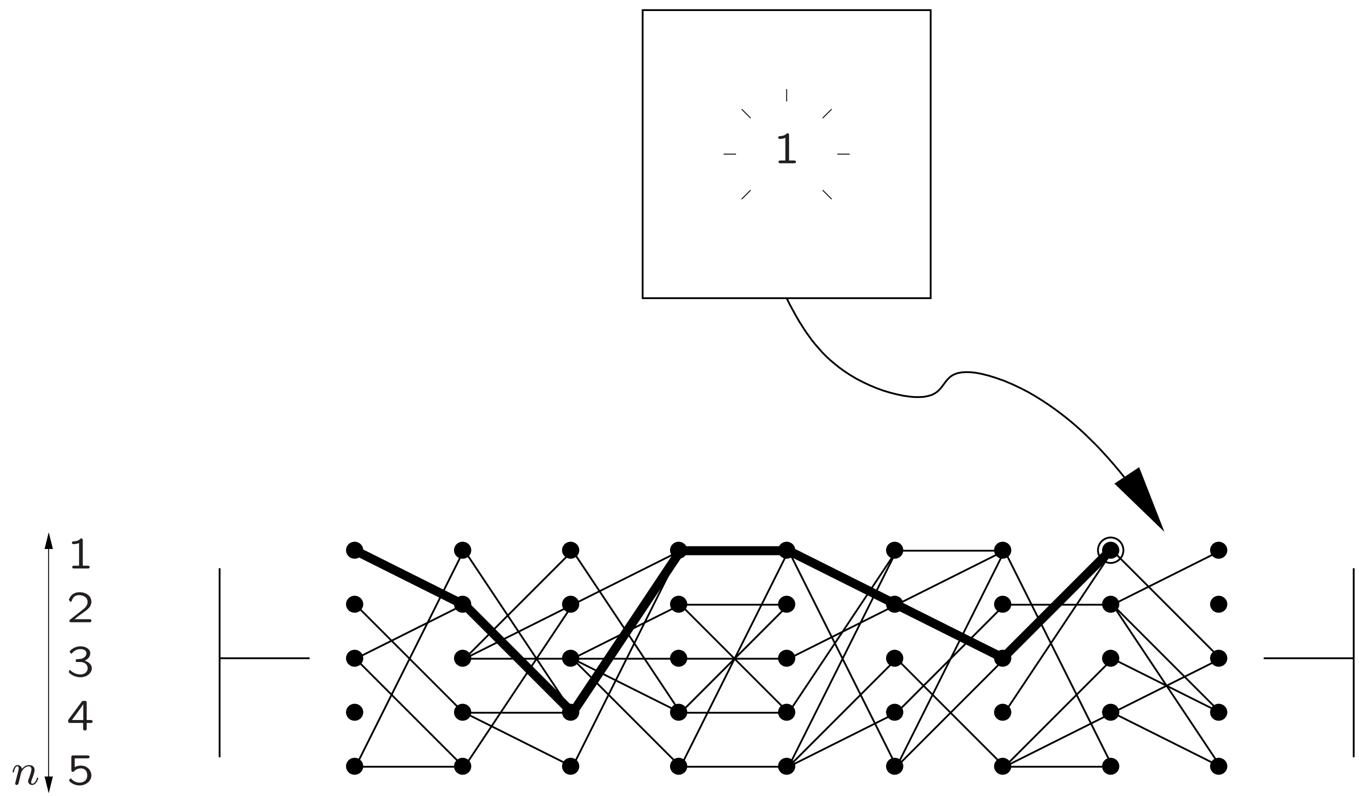
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



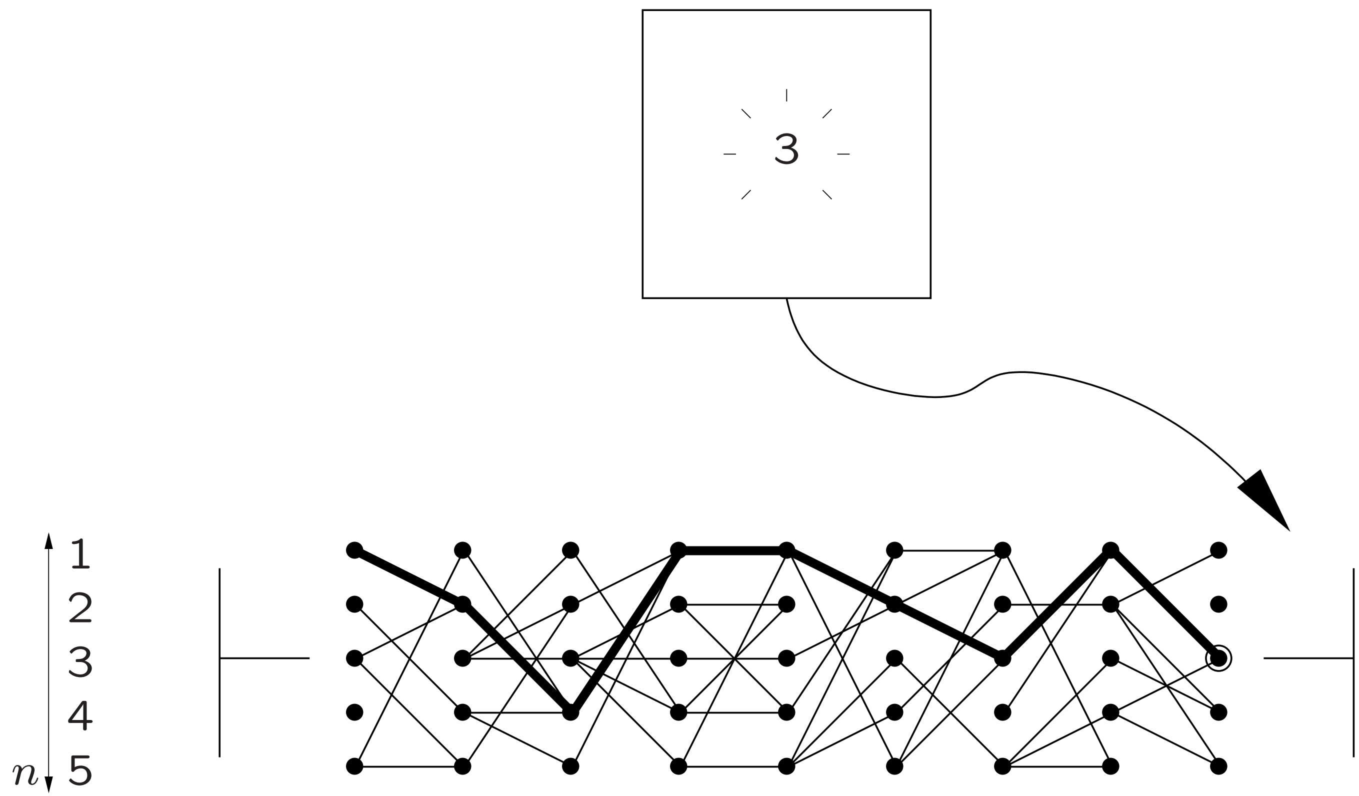
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



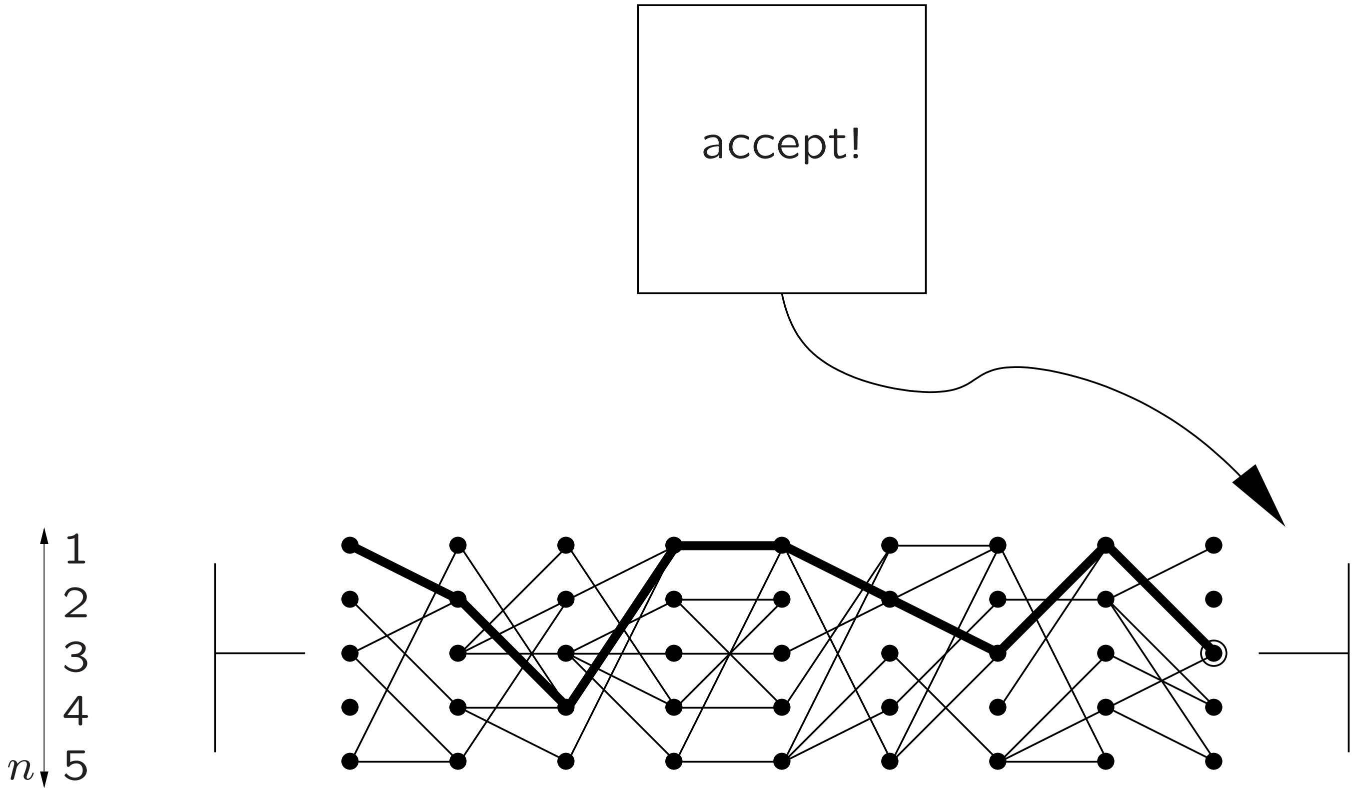
is there a *live* path?

(input alphabet has 2^{n^2} symbols)



is there a *live* path?

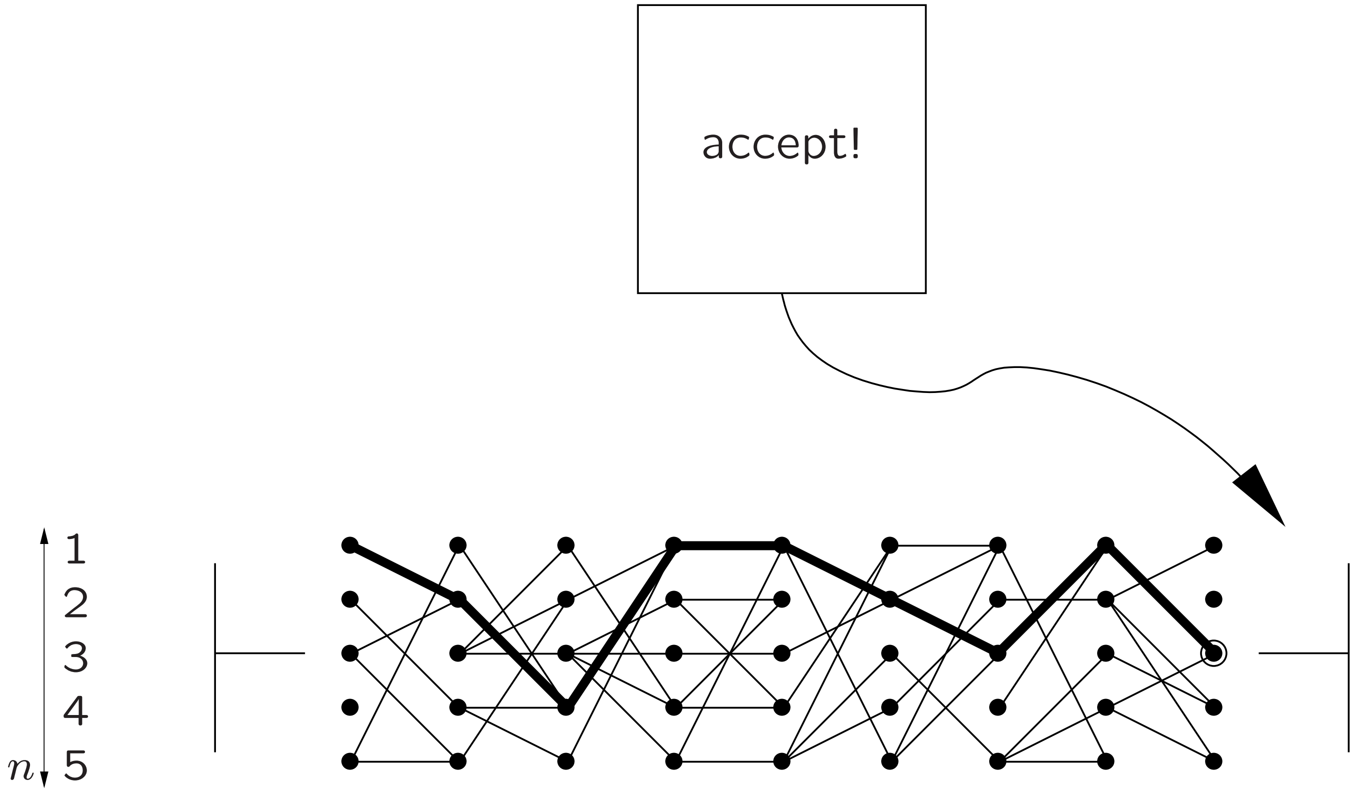
(input alphabet has 2^{n^2} symbols)



is there a *live* path?

(input alphabet has 2^{n^2} symbols)

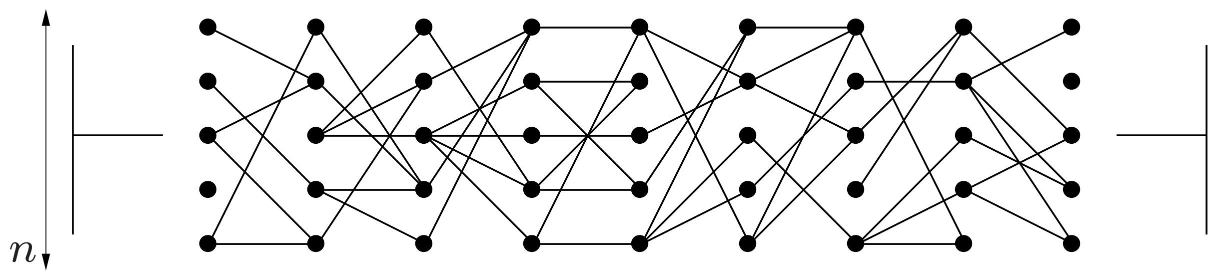
complement of one-way liveness

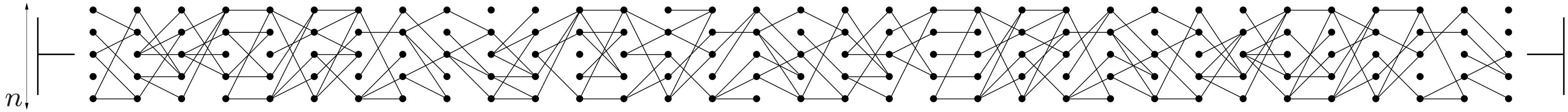


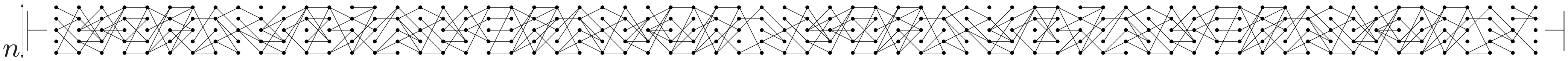
is there no *live* path?

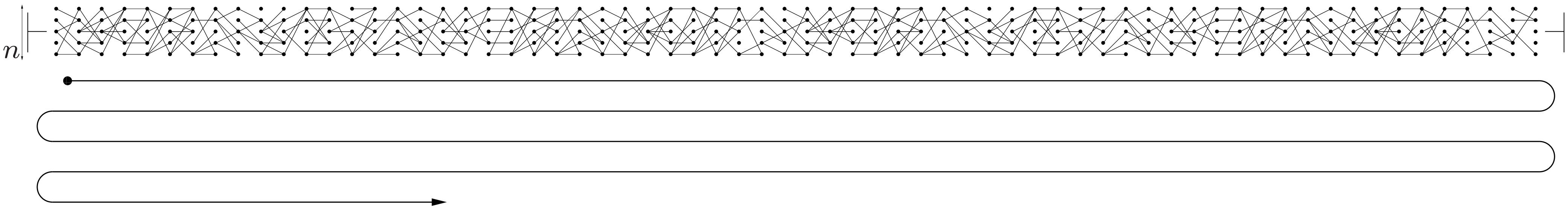
(input alphabet has 2^{n^2} symbols)

proof outline


n







PROOF

Suppose some k-state sweeping 2NFA S solves the complement of liveness.

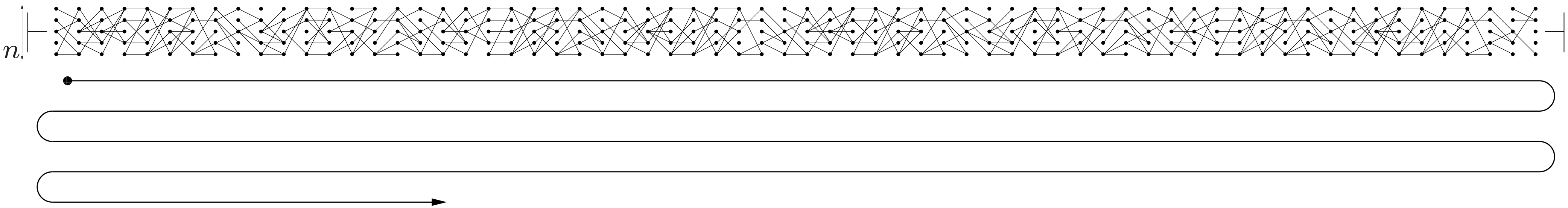
We will construct $N \times N$ "hard" inputs, where $N ::= (2^n - 1)^2$.

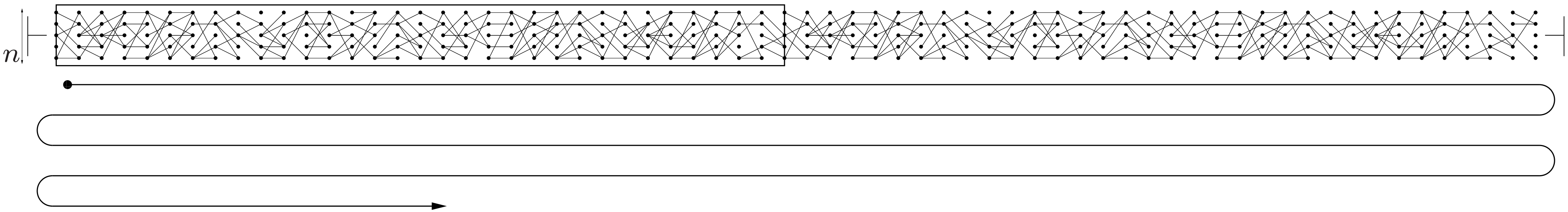
S behaves "appropriately" on all these inputs $\implies k^2 + \binom{k^2}{2} \geq N$

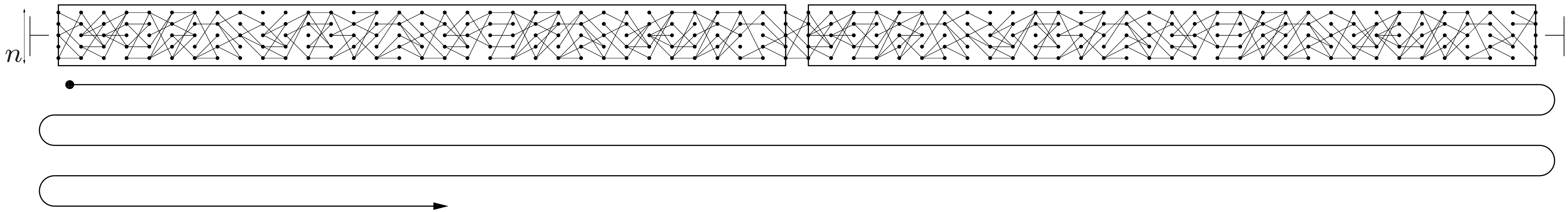
Therefore $k = 2^{\Omega(n)}$.

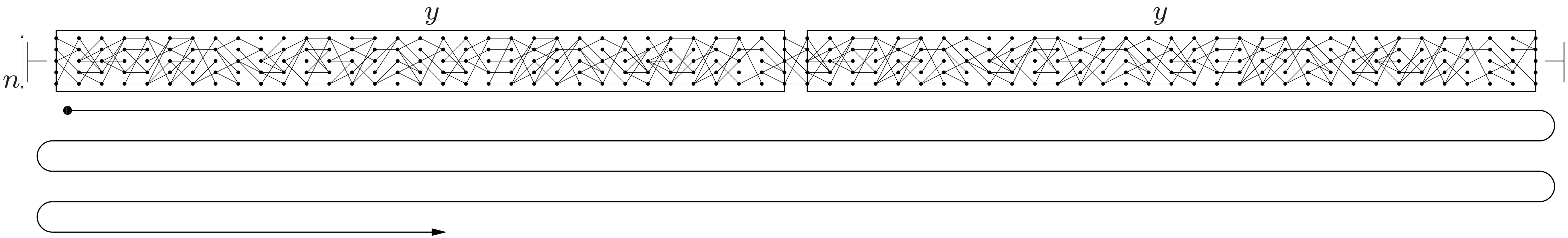
QED

the hard inputs

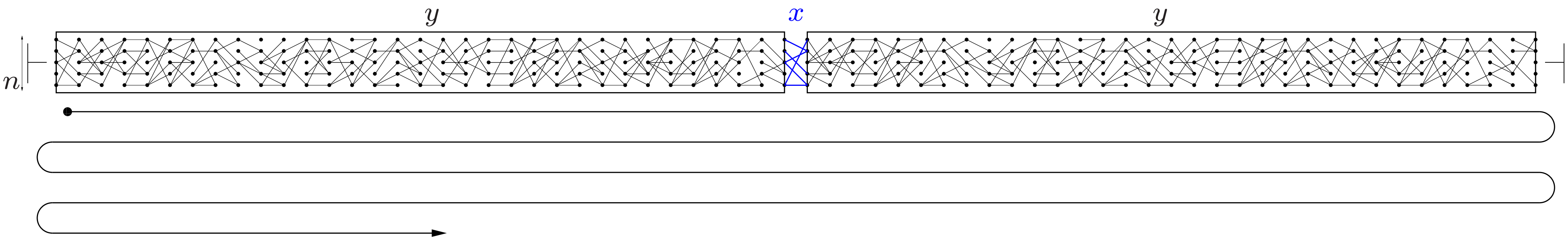


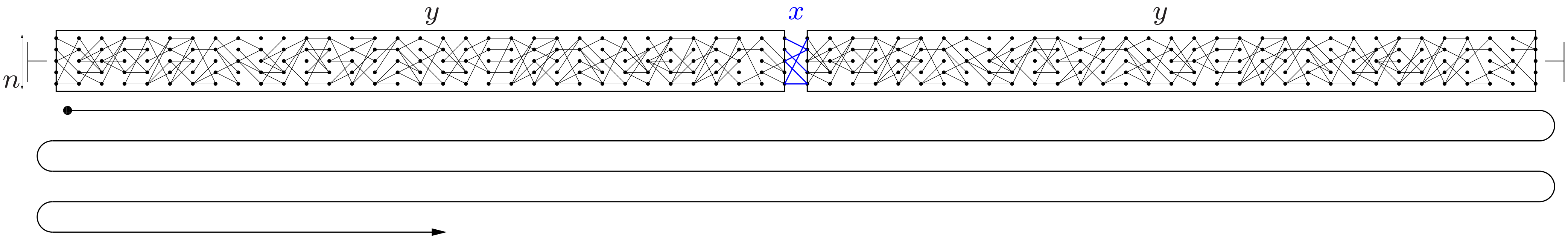

n


n


y
y
n

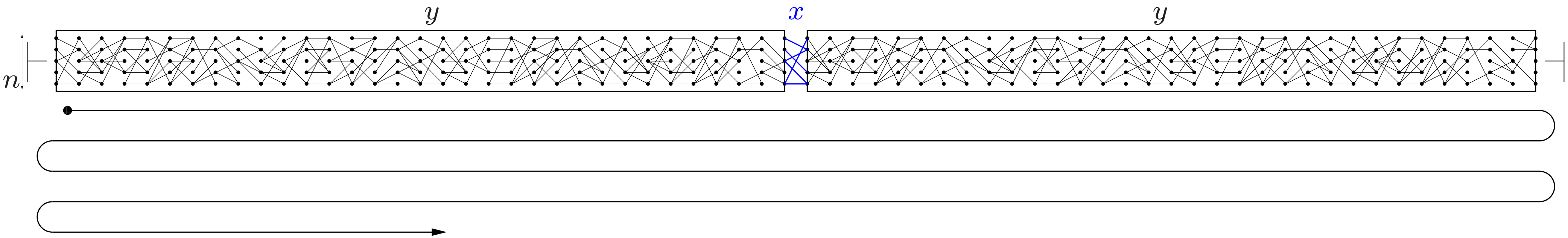
the hard inputs





STEP 1

- find a y that "exhausts" the machine in either direction
- check the machine's behavior on yxy for any x from a list $x_1, x_2, \ldots, x_N$



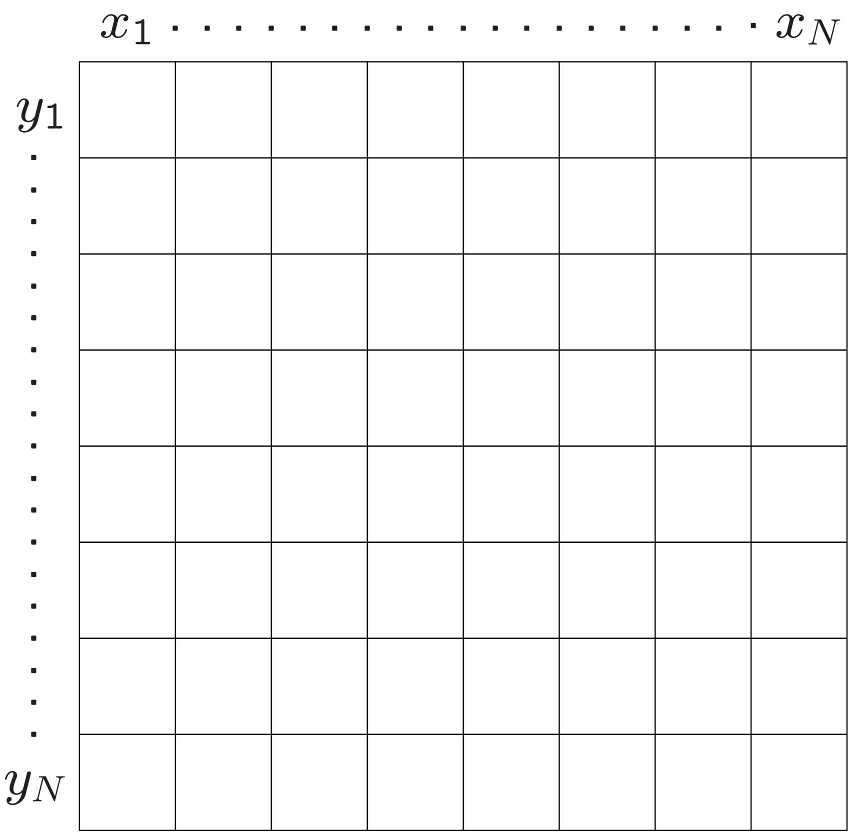
STEP 1

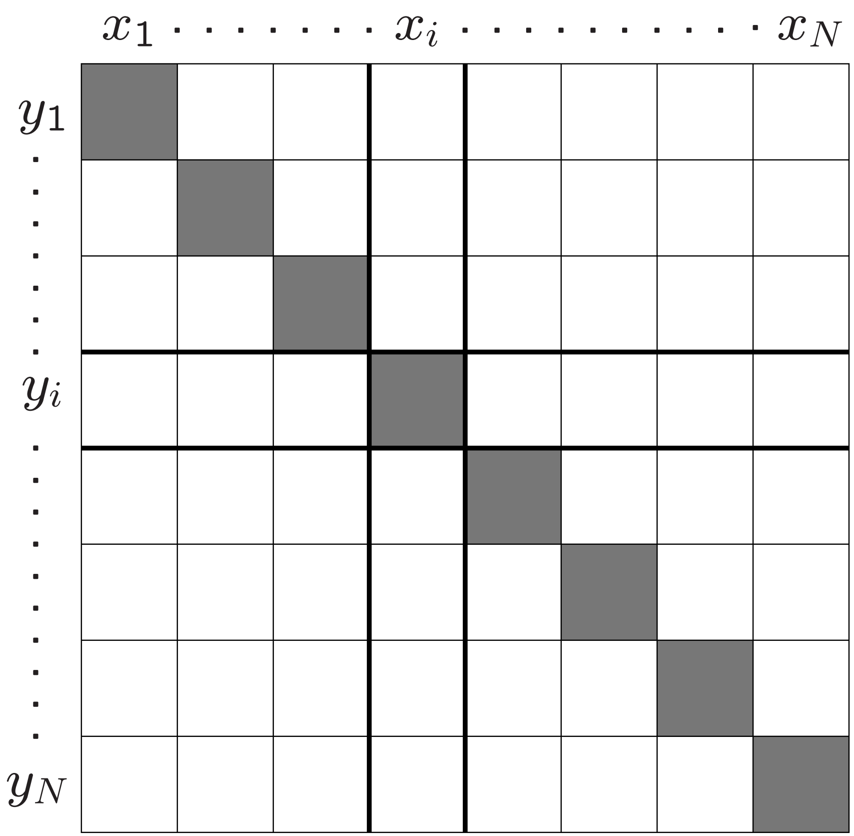
- find a y that "exhausts" the machine in either direction
- check the machine's behavior on yxy for any x from a list $x_1, x_2, \ldots, x_N$

STEP 2

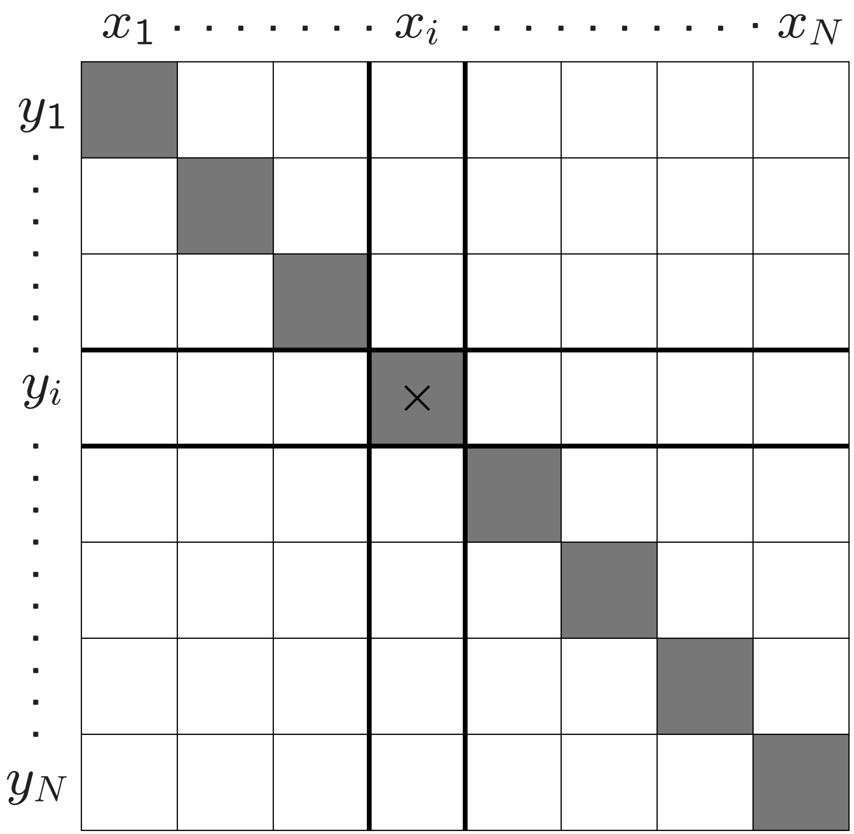
- repeat STEP 1 for any y from a list of "exhausting" strings $y_1, y_2, \ldots, y_N$

the hard inputs

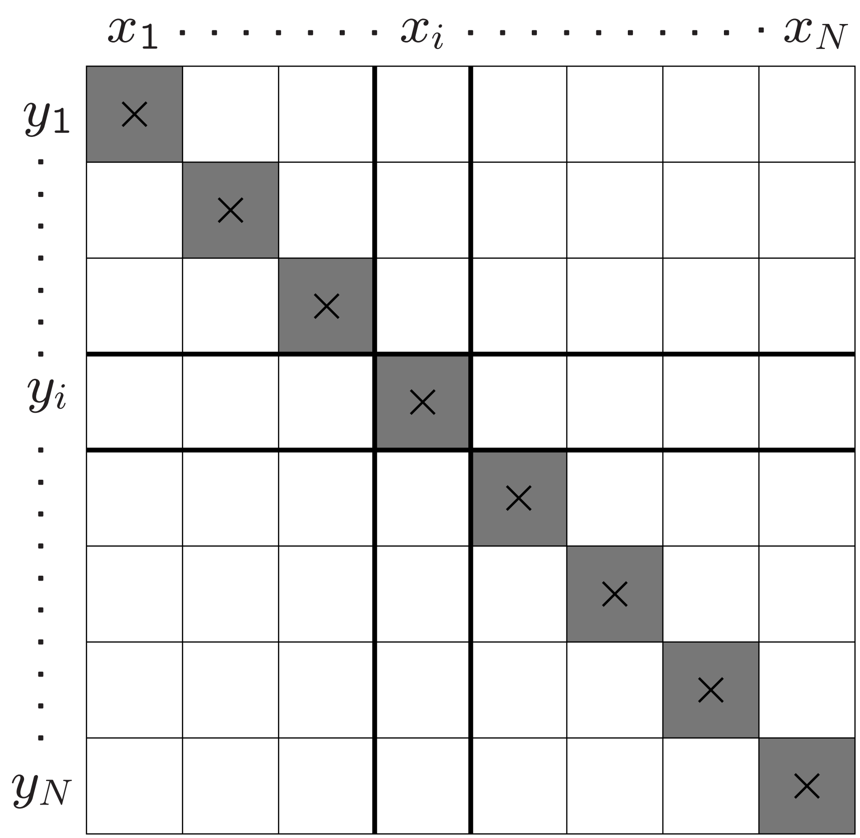



x_1 · · · · · · · x_i · · · · · · · · · x_N
y_1
y_i
y_N

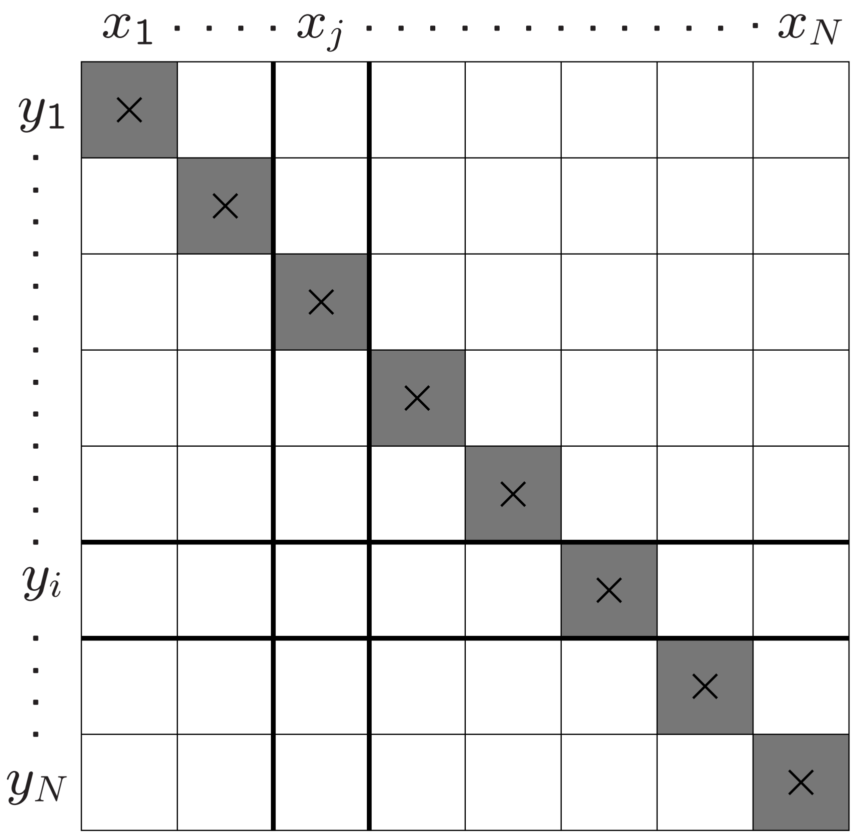
the hard inputs

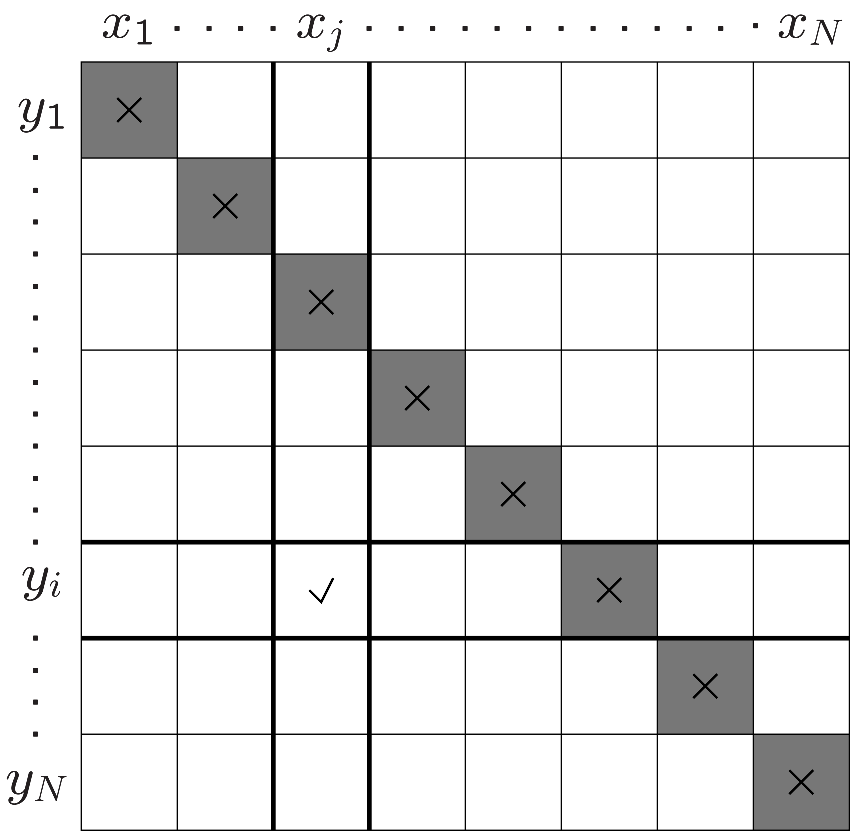


the hard inputs



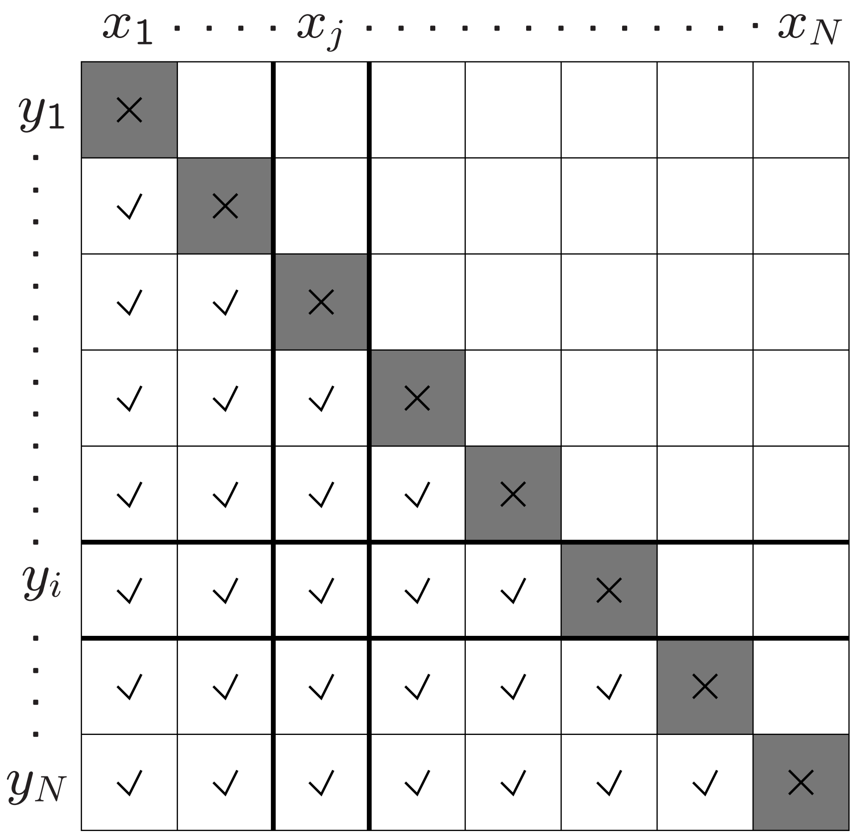
the hard inputs



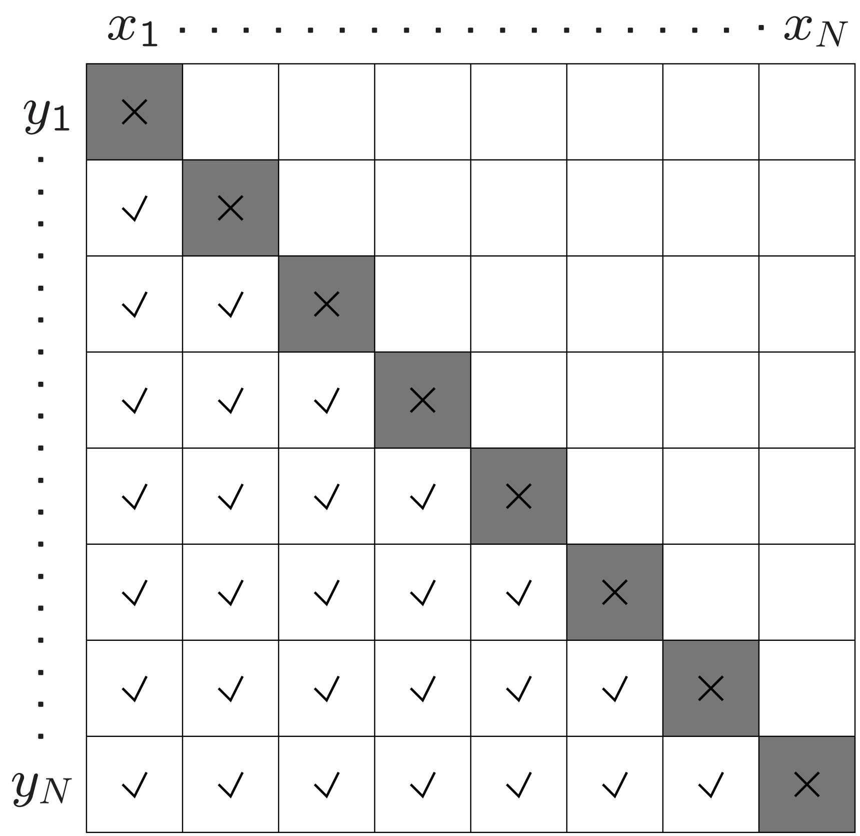


x_1
x_j
x_N
y_1
y_i
y_N

the hard inputs



the hard inputs

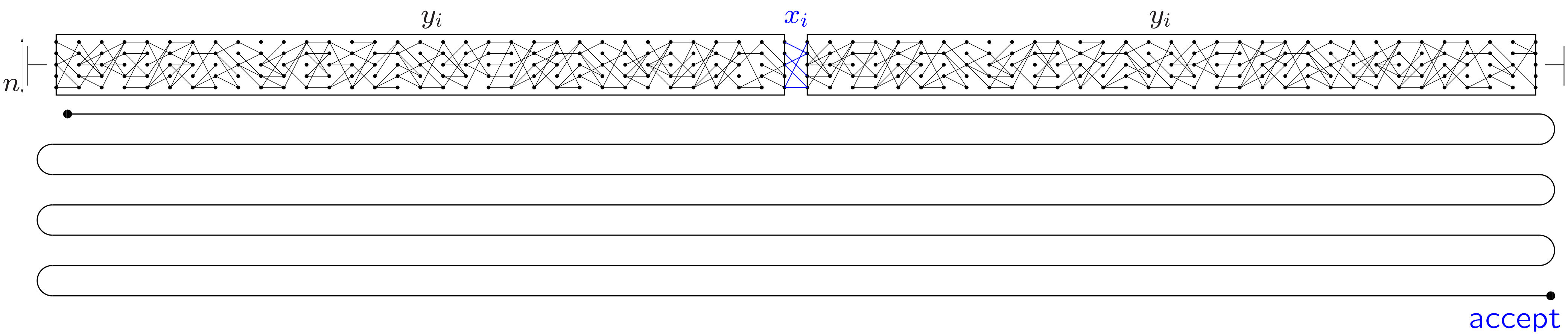


the hard inputs

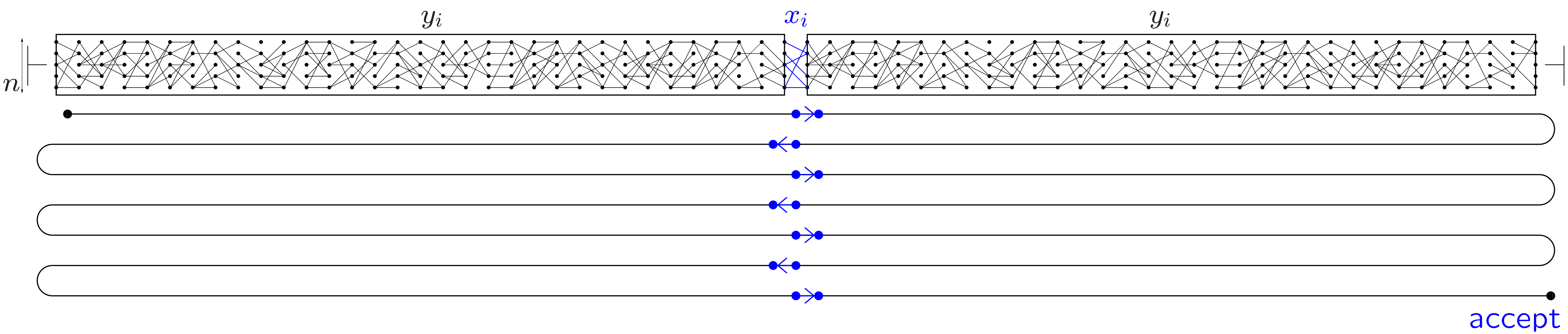
	x_1							x_N
y_1	yes							
	no	yes						
	no	no	yes					
	no	no	no	yes				
	no	no	no	no	yes			
	no	no	no	no	no	yes		
	no	no	no	no	no	no	yes	
y_N	no	no	no	no	no	no	no	yes

	x_1			x_i				x_N
y_1	yes							
	no	yes						
	no	no	yes					
y_i	no	no	no	yes				
	no	no	no	no	yes			
	no	no	no	no	no	yes		
	no	no	no	no	no	no	yes	
y_N	no	no	no	no	no	no	no	yes

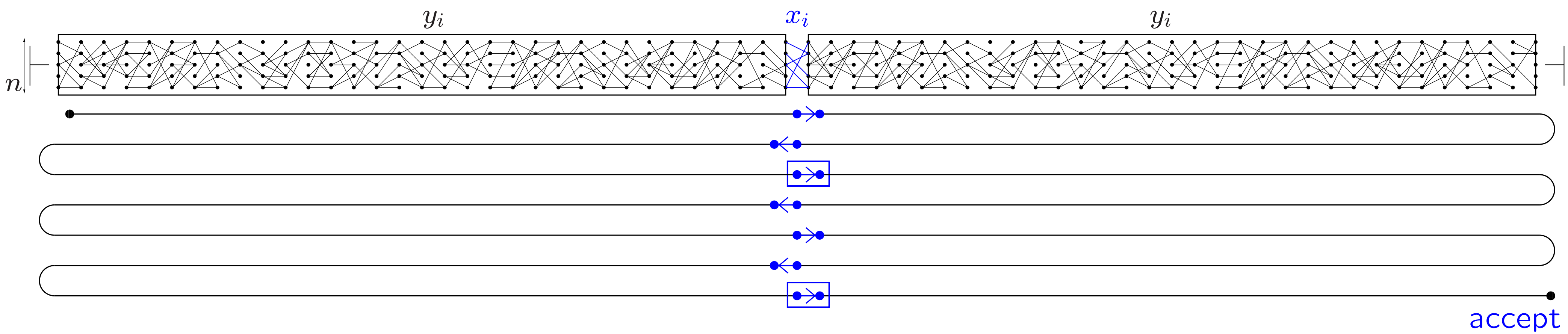
the hard inputs

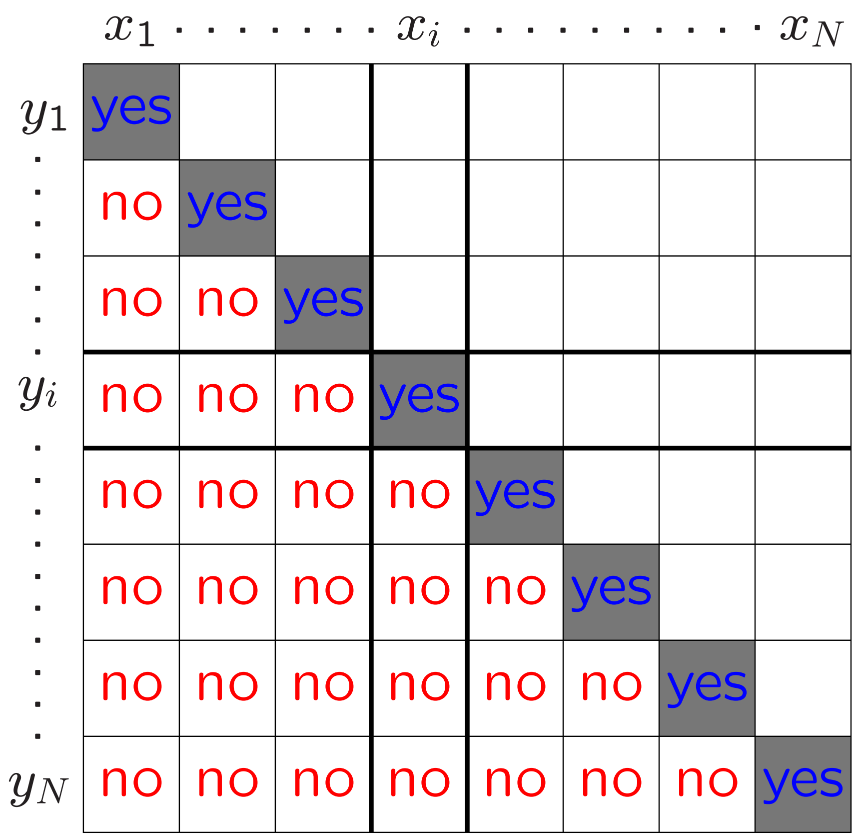


the hard inputs



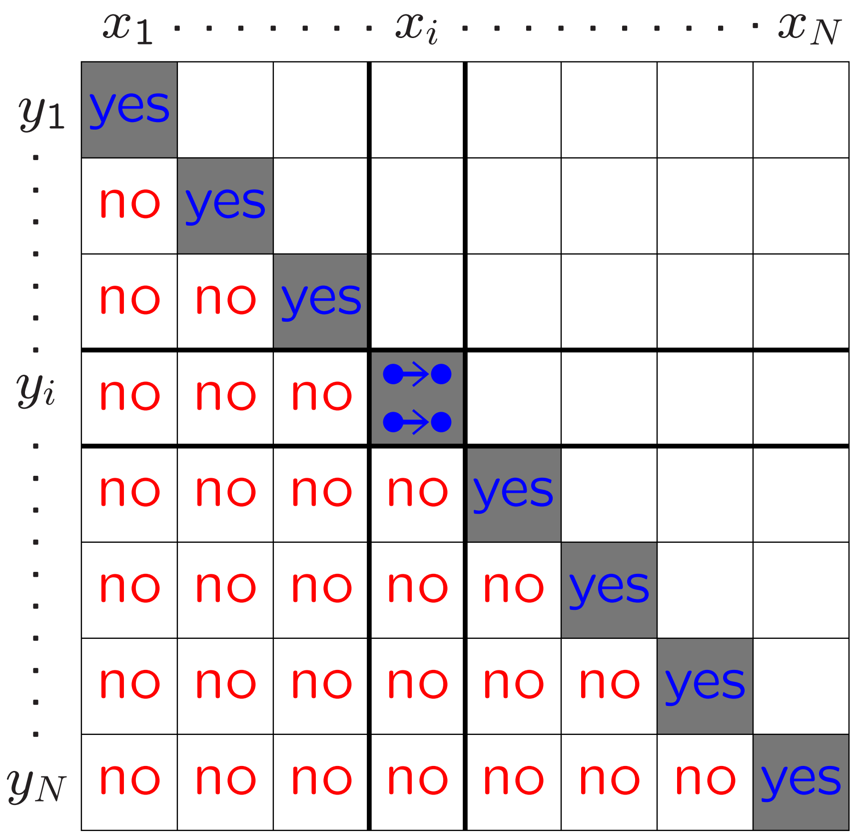
the hard inputs

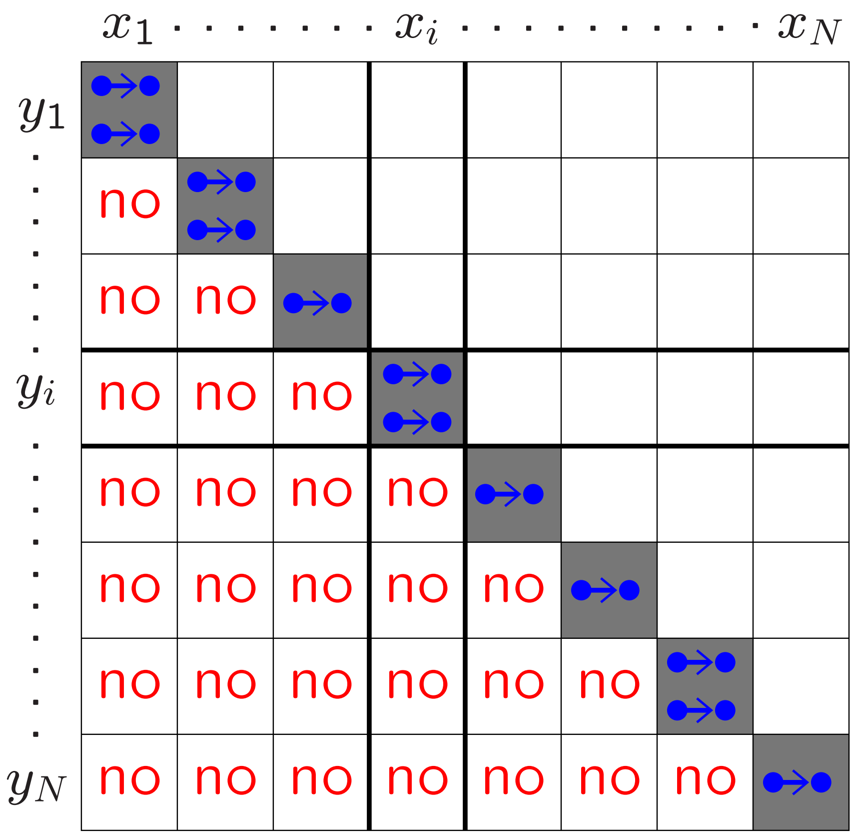




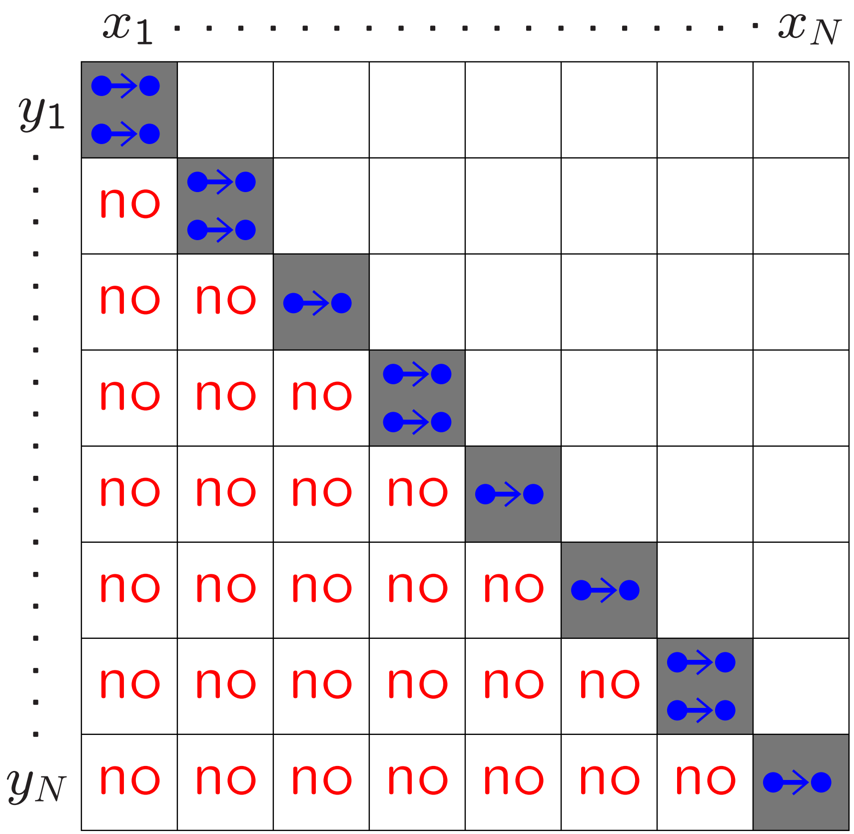
x_1 x_i x_N
y_1
y_i
y_N
yes
no yes
no no yes
no no no yes
no no no no yes
no no no no no yes
no no no no no no yes
no no no no no no no yes

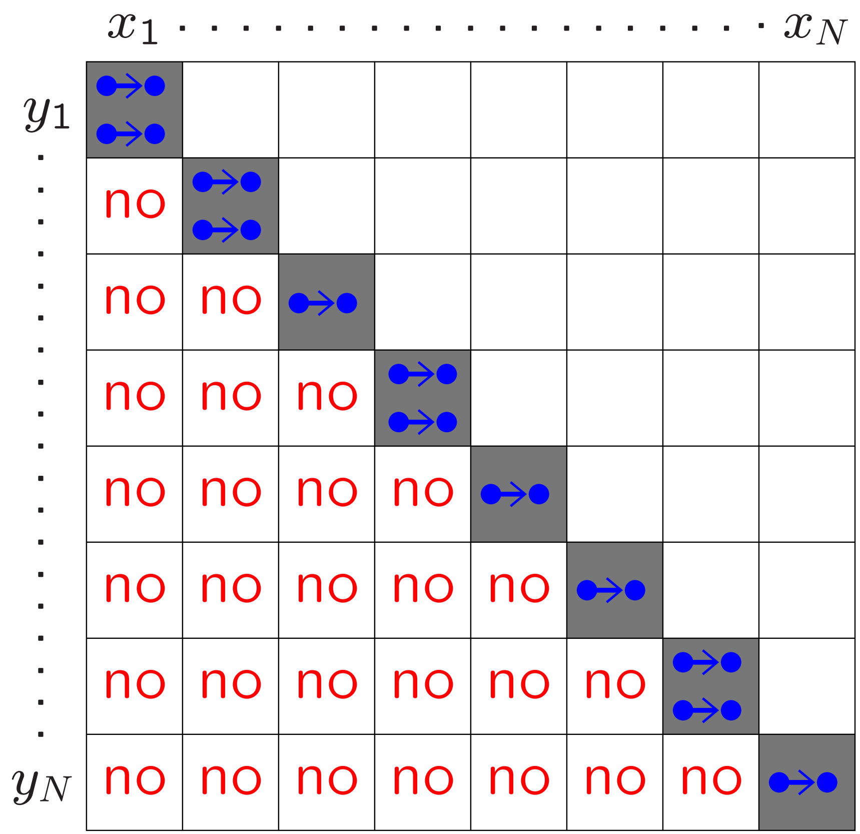
the hard inputs




x_1 x_i x_N
y_1 y_i y_N
no

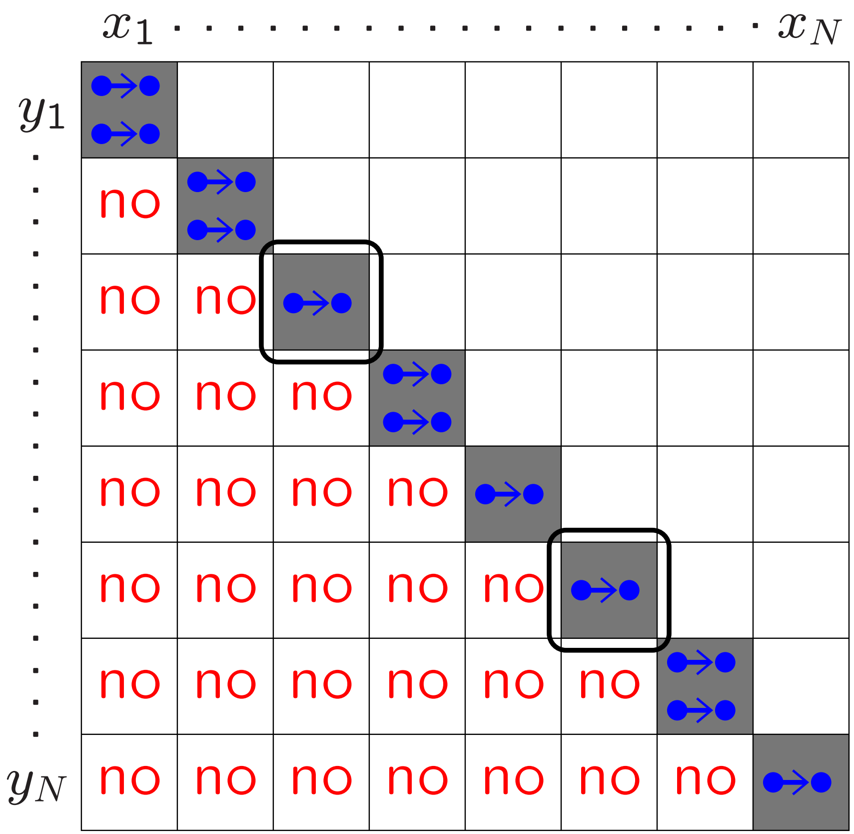
the hard inputs



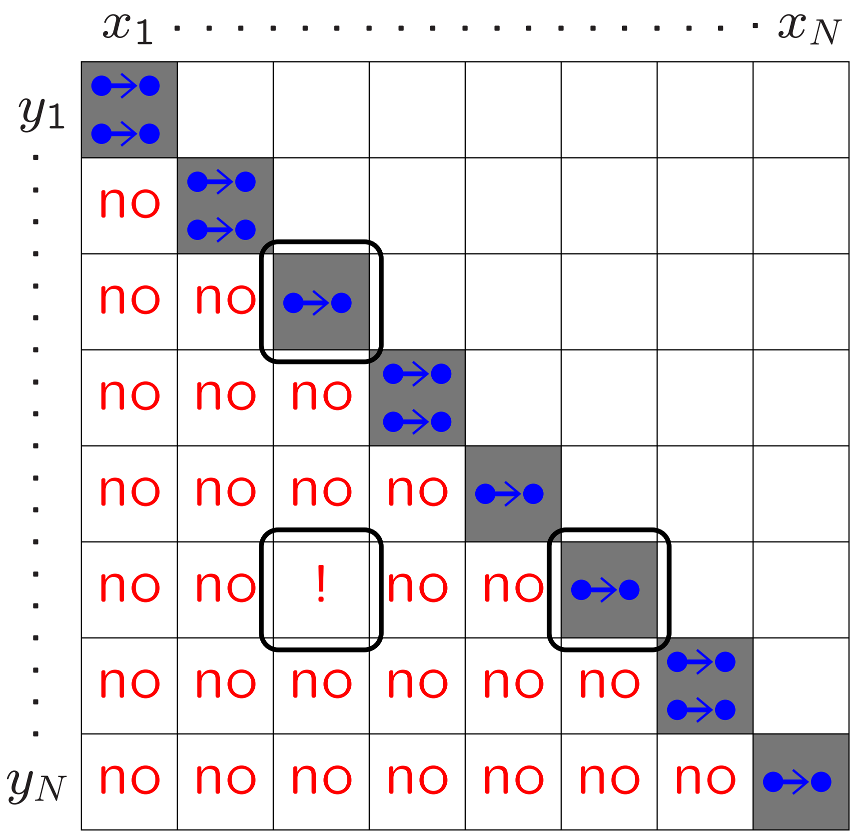


diagonal cells must have distinct contents!

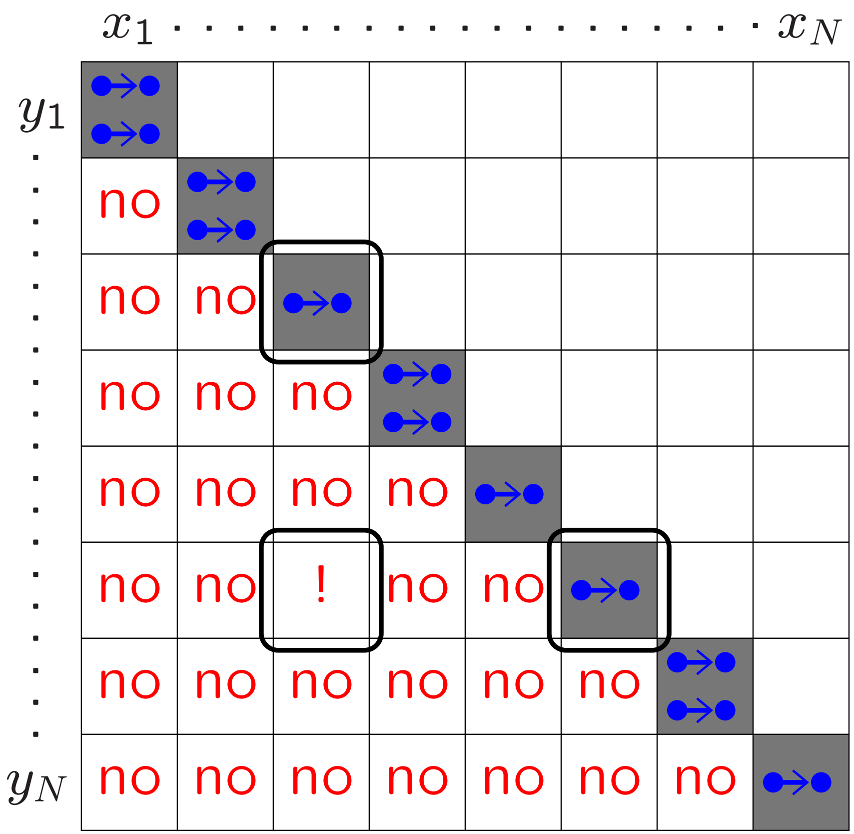
the hard inputs



diagonal cells must have distinct contents!

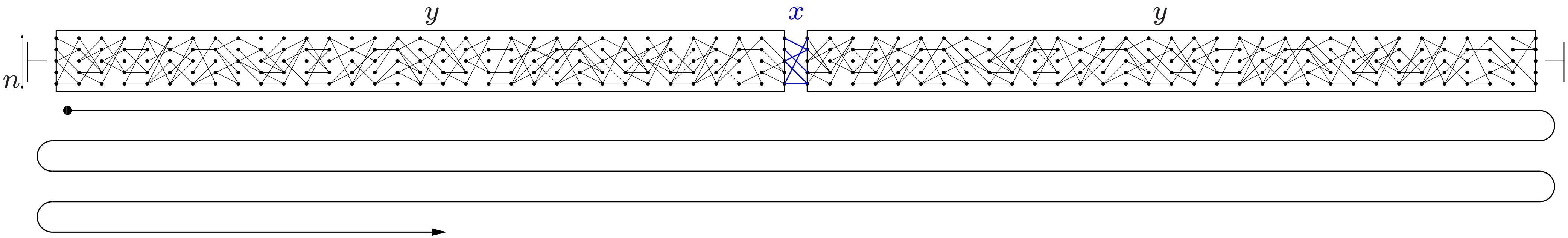


diagonal cells must have distinct contents!



diagonal cells must have distinct contents!

$$k^2 + \binom{k^2}{2} \geq N$$



PROOF

Suppose some k-state sweeping 2NFA S solves the complement of liveness.

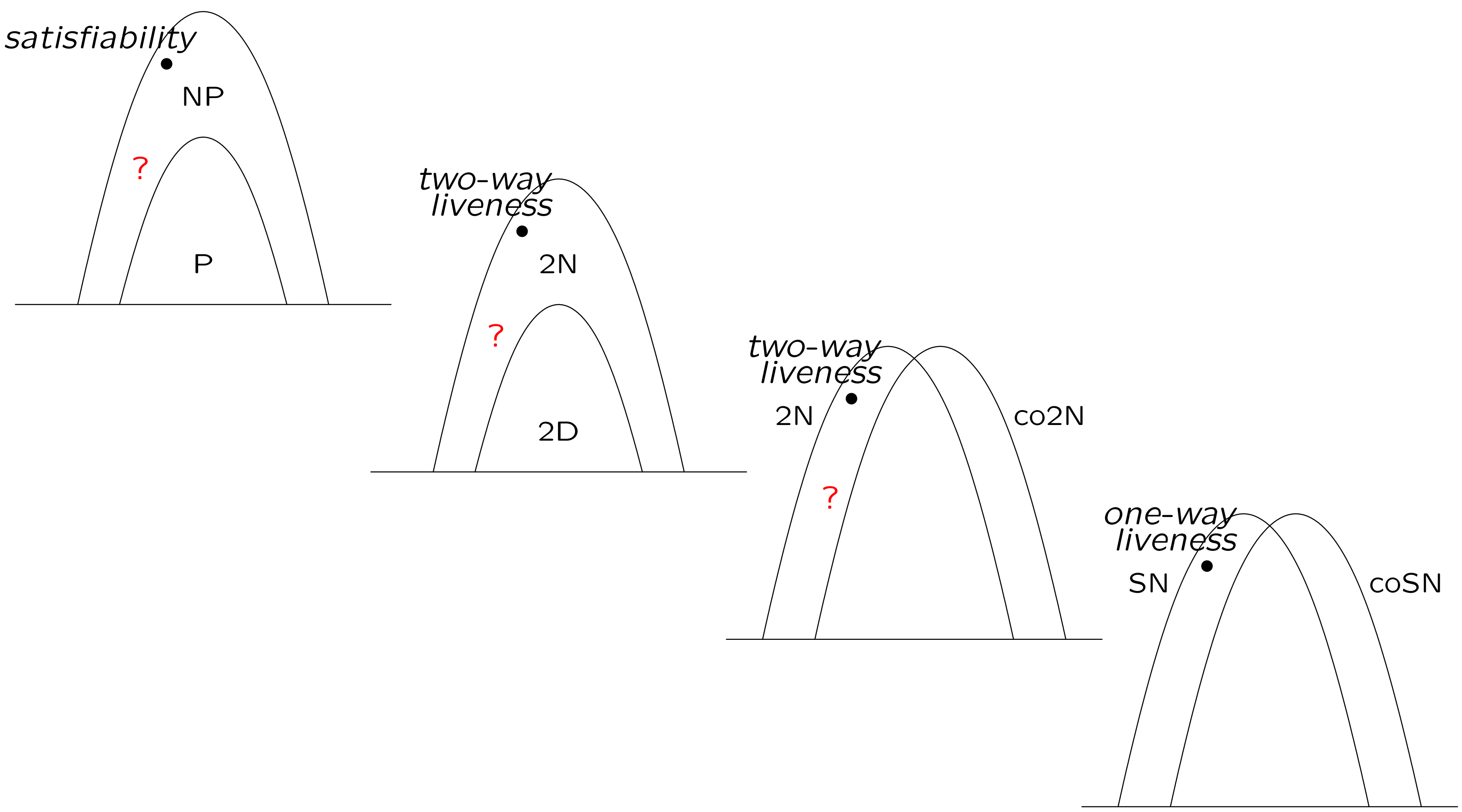
We will construct $N \times N$ "hard" inputs, where $N ::= (2^n - 1)^2$.

S behaves "appropriately" on all these inputs $\implies k^2 + \binom{k^2}{2} \geq N$

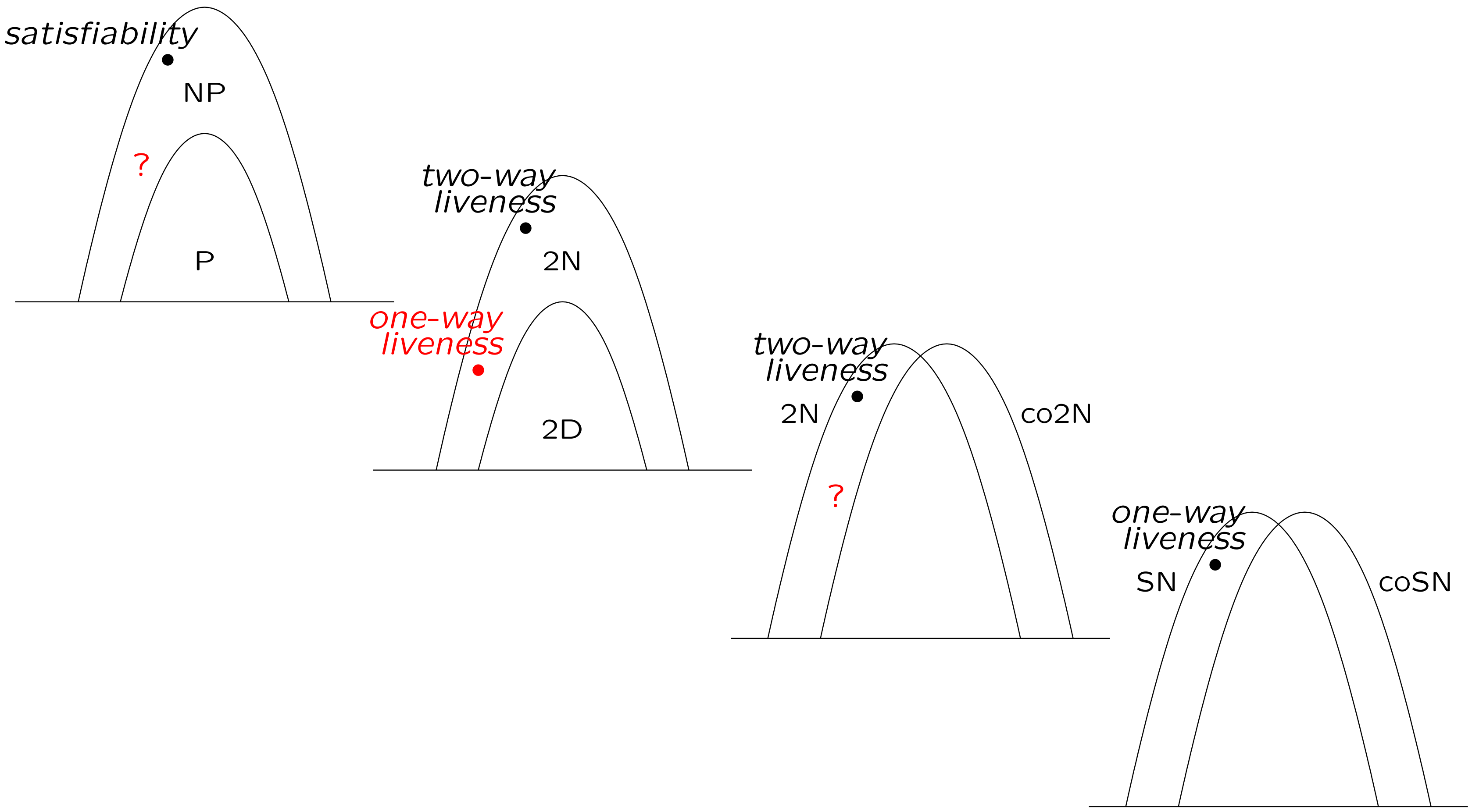
Therefore $k = 2^{\Omega(n)}$.

QED

conclusion



no small *sweeping* 2NFA can solve the *complement* of one-way liveness



can a small 2DFA solve one-way liveness?