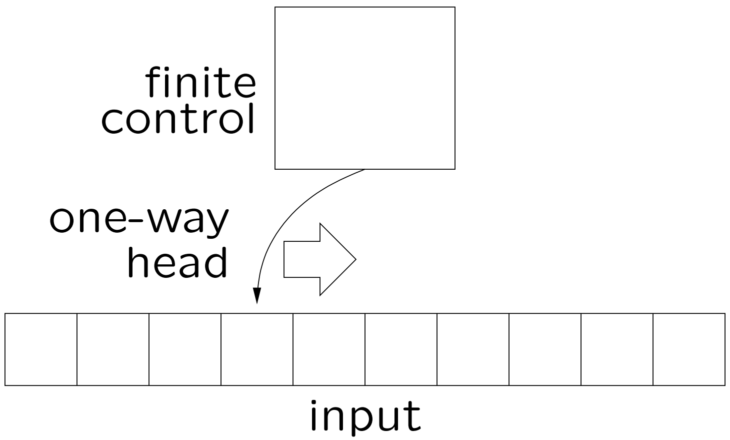
Deterministic moles cannot solve liveness

Christos Kapoutsis

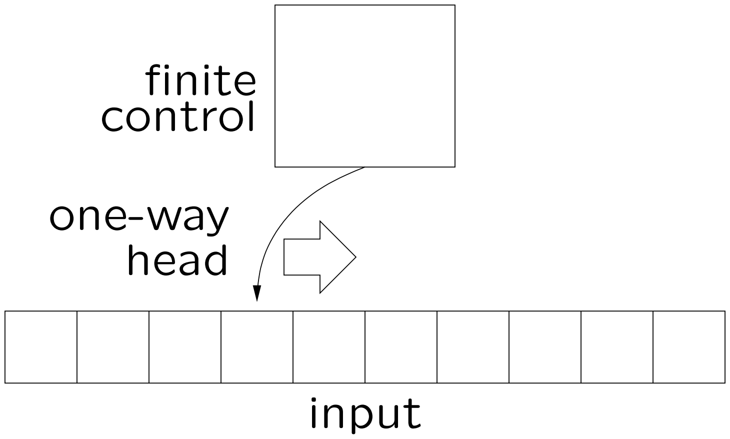
workshop on the
Descriptional Complexity of Formal Systems
Como, Italy, July 2005

1DFA

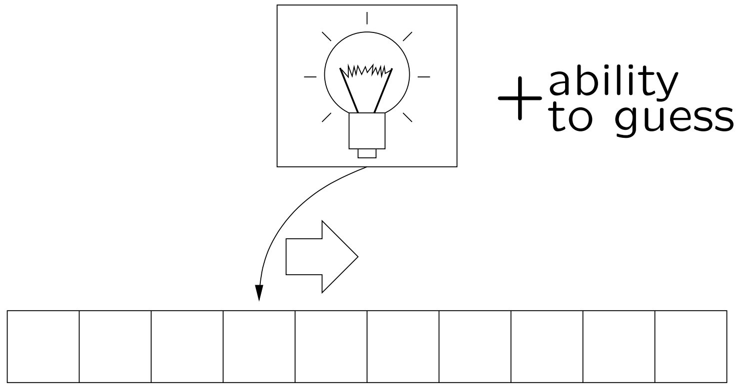


1DFAs versus 1NFAs

1DFA

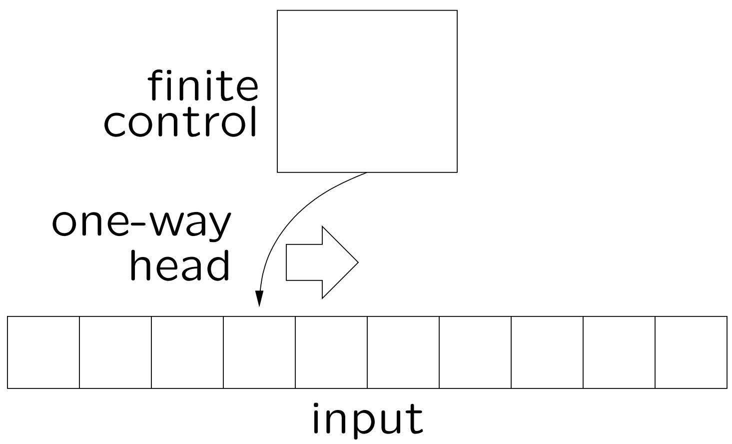


1NFA

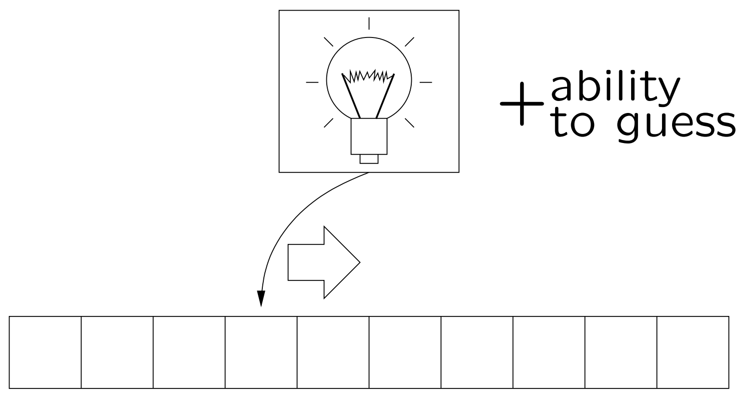


1DFAs versus 1NFAs

1DFA



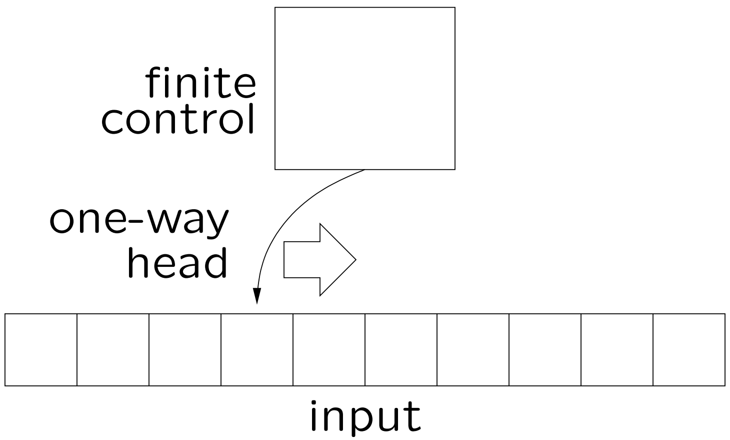
1NFA



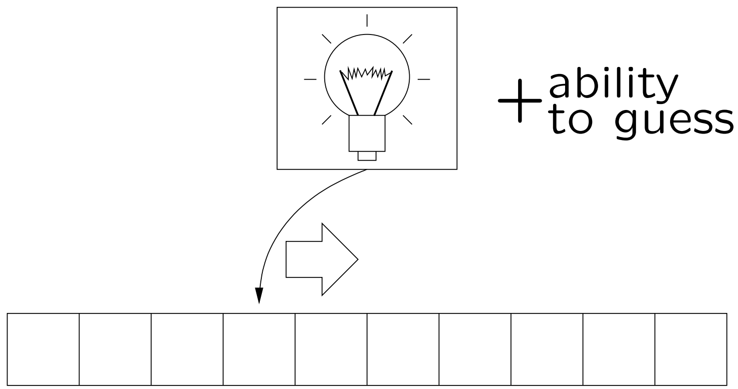
a 1DFA with $\leq 2^n - 1$ states $\xleftarrow{\text{can be converted to}}$ every 1NFA with n states

1DFAs versus 1NFAs

1DFA



1NFA

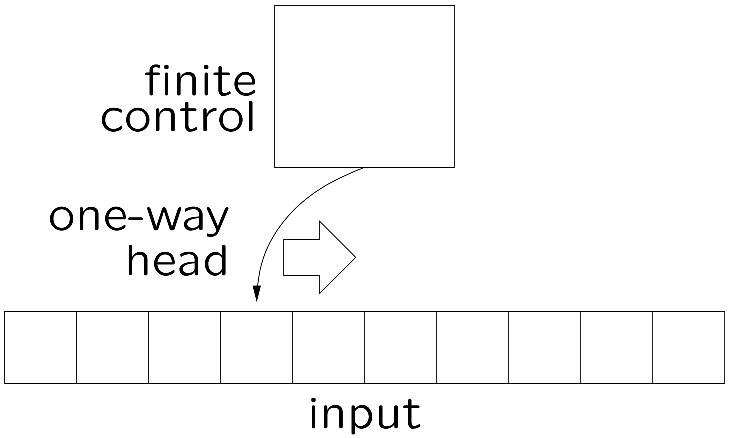


a 1DFA with $\leq 2^n - 1$ states $\xleftarrow{\text{can be converted to}}$ every 1NFA with n states

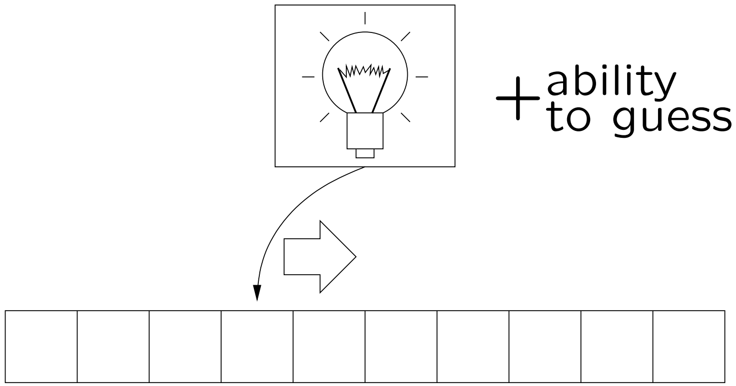
and sometimes all these $2^n - 1$ states are necessary

1DFAs versus 1NFAs

1DFA



1NFA

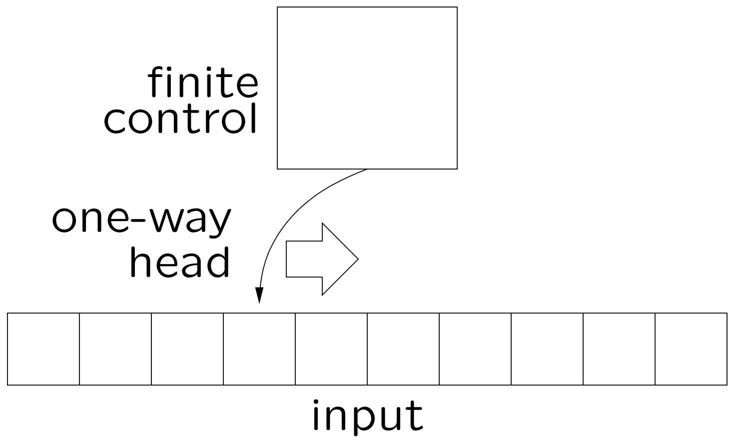


a 1DFA with $\leq 2^n - 1$ states $\xleftarrow{\text{can be converted to}}$ every 1NFA with n states

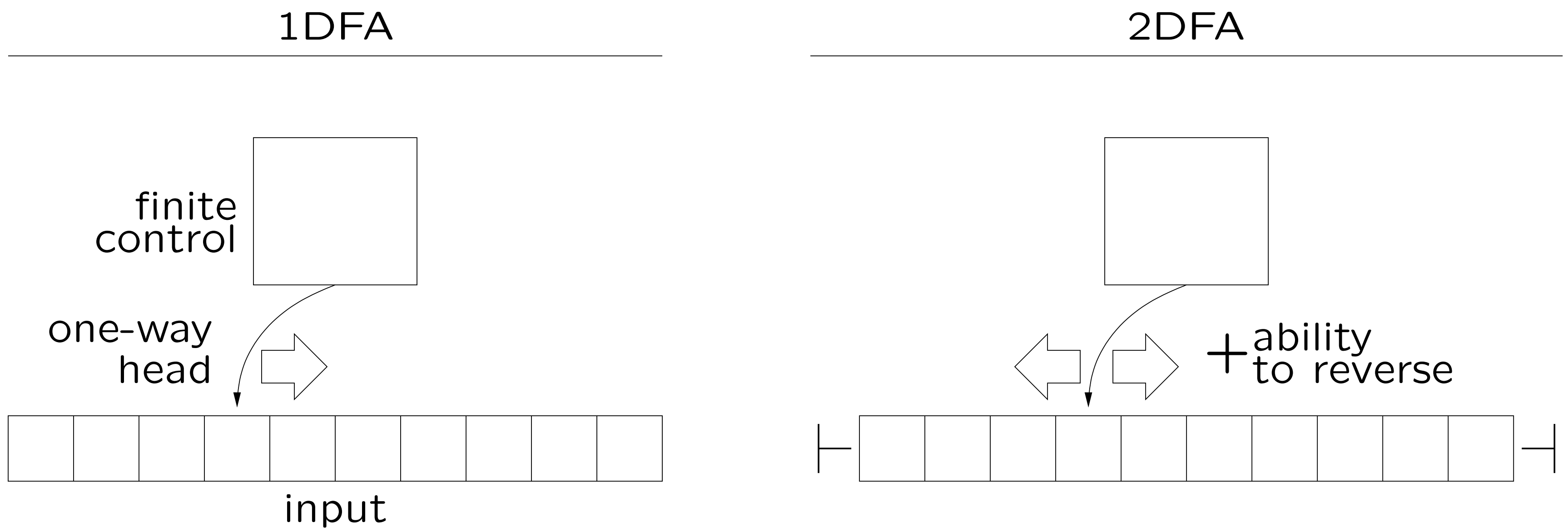
and sometimes all these $2^n - 1$ states are necessary

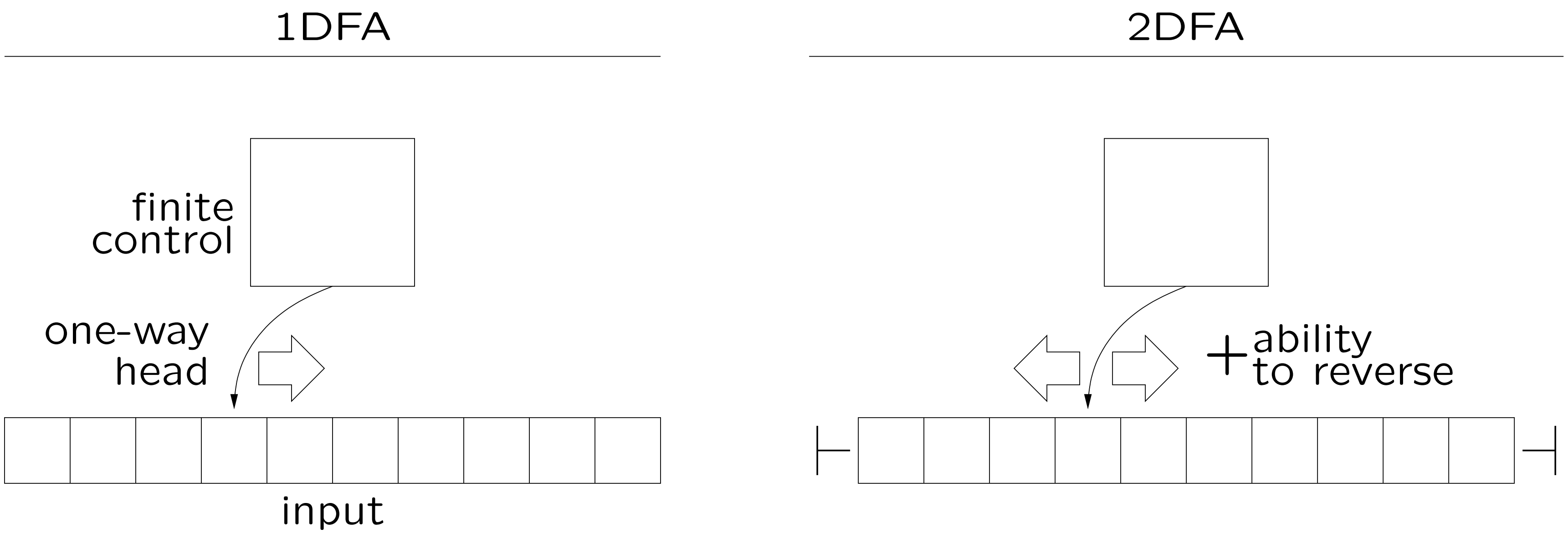
"*the trade-off is exactly* $2^n - 1$"

1DFA



1DFAs versus 2DFAs



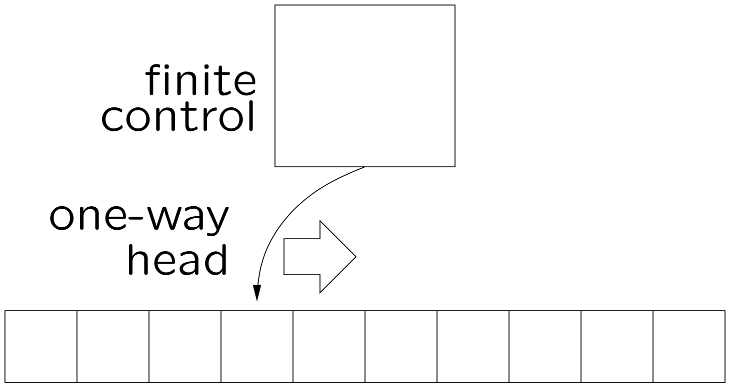


a 1DFA with $\leq$ $\boxed{\ ?\ }$ states $\xleftarrow{\text{can be converted to}}$ every 2DFA with n states

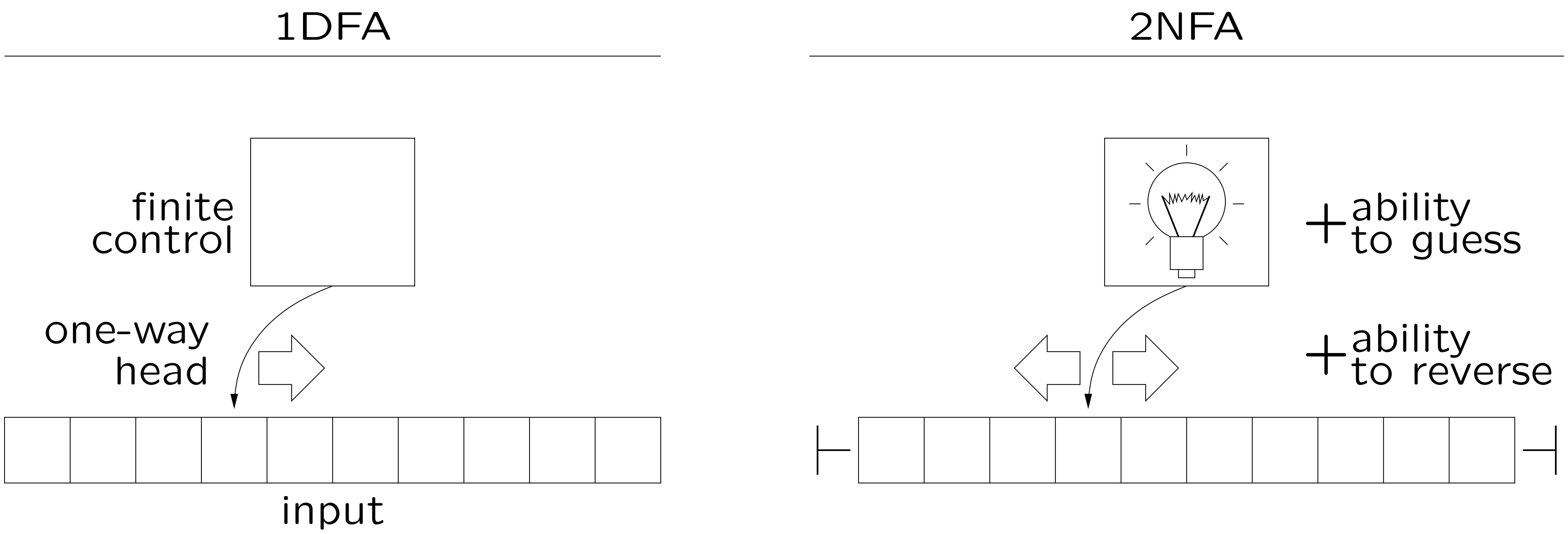
and sometimes all these $\boxed{\ ?\ }$ states are necessary

"*the trade-off is exactly* $\boxed{\ ?\ }$ "

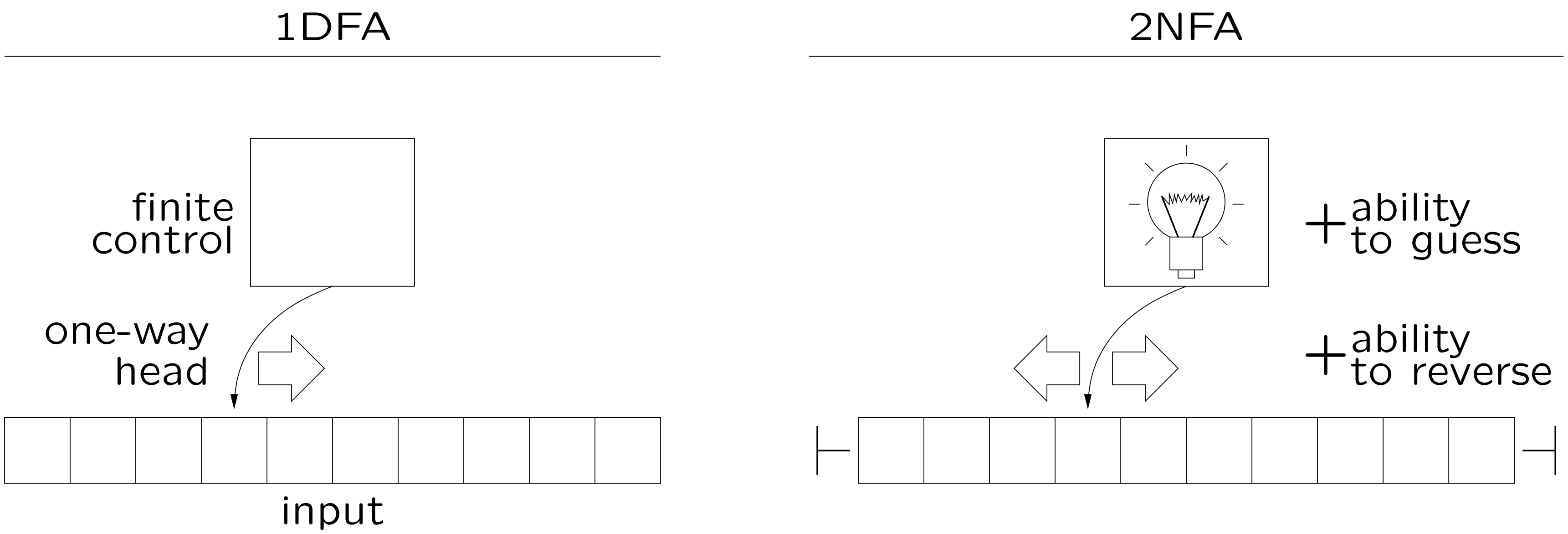
1DFA



1DFAs versus 2NFAs



1DFAs versus 2NFAs

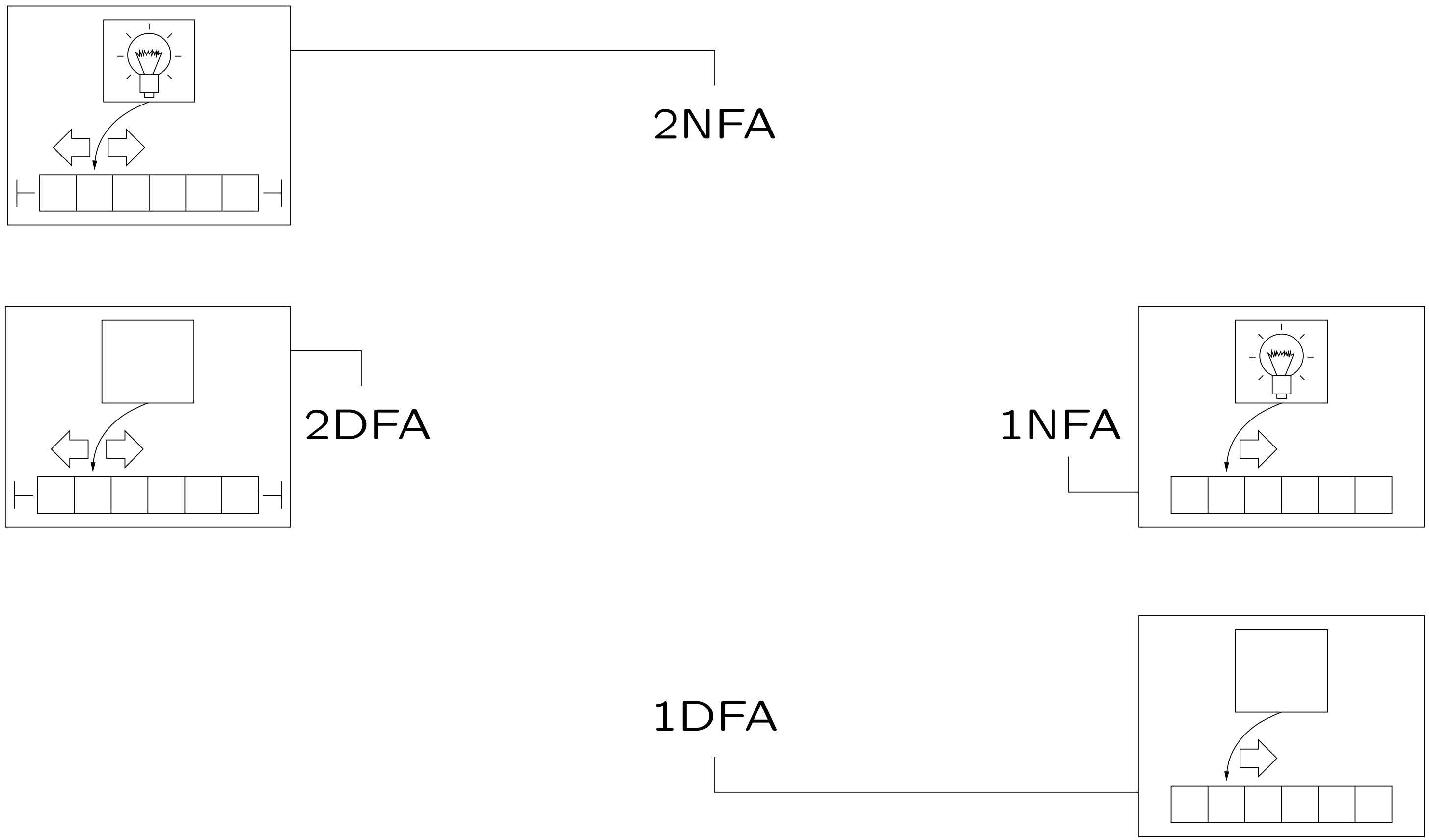


a 1DFA with $\leq$ $\boxed{?}$ states $\xleftarrow{\text{can be converted to}}$ every 2NFA with n states

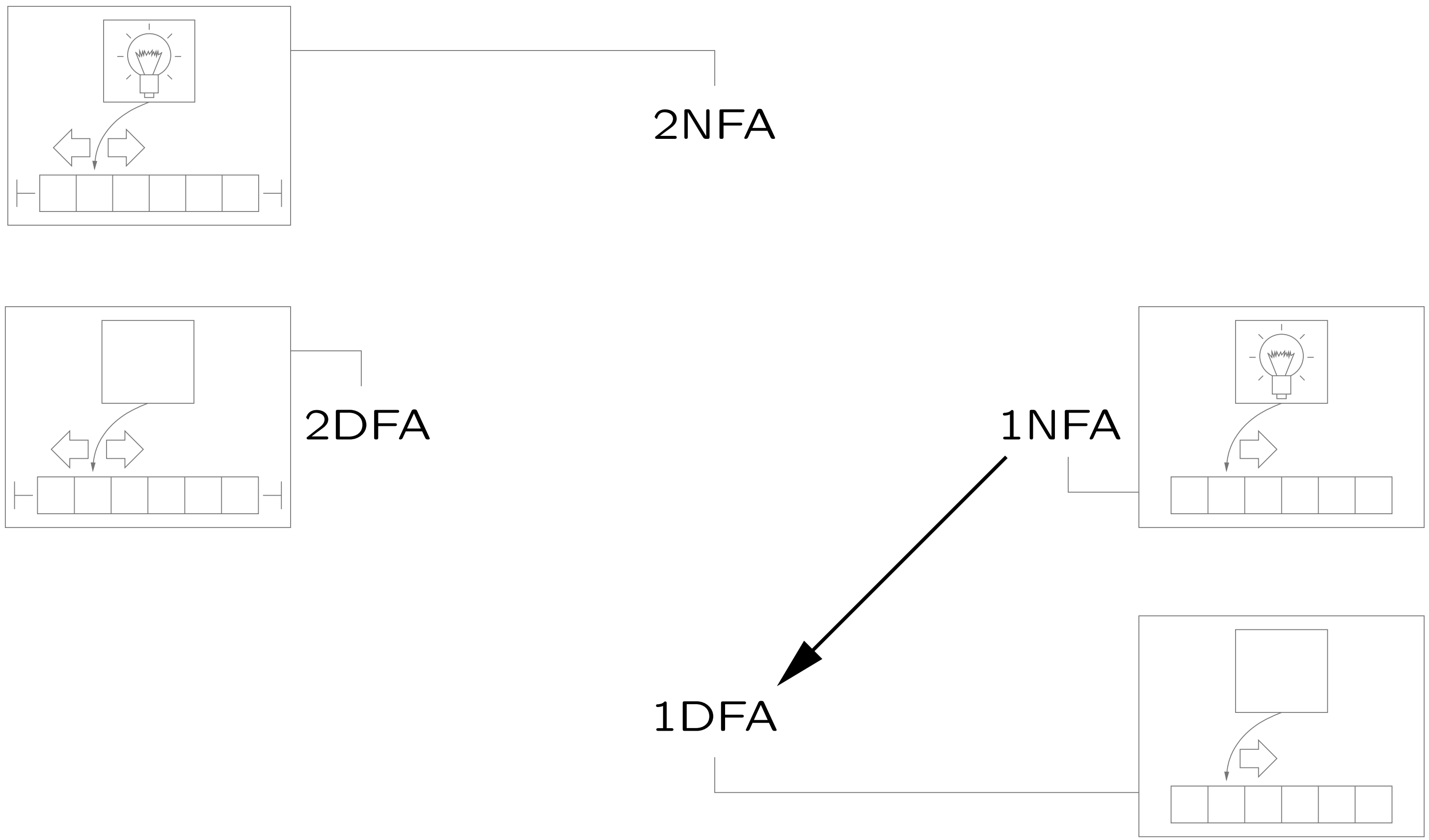
and sometimes all these $\boxed{?}$ states are necessary

"*the trade-off is exactly* $\boxed{?}$ "

the big picture

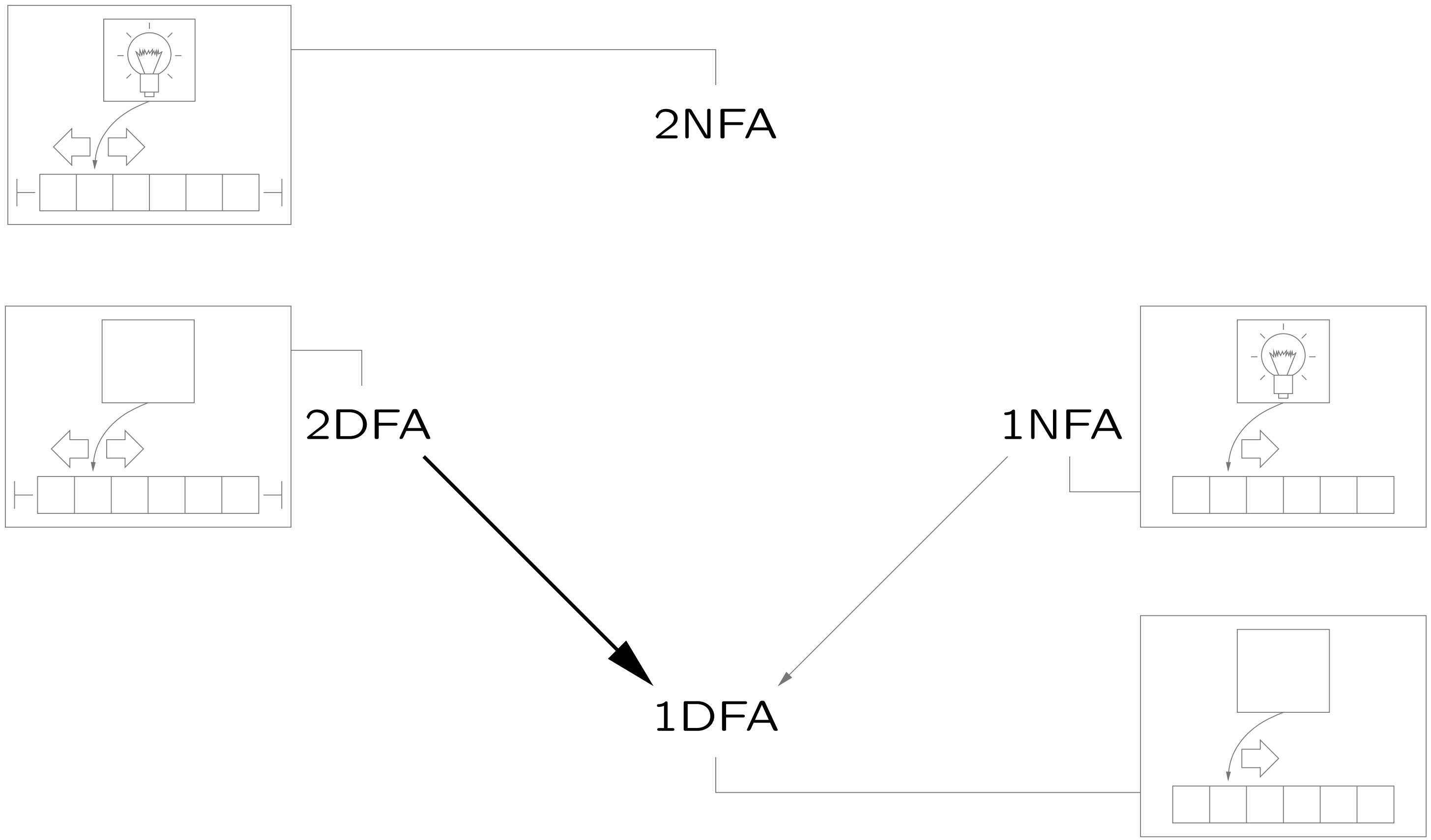


the big picture



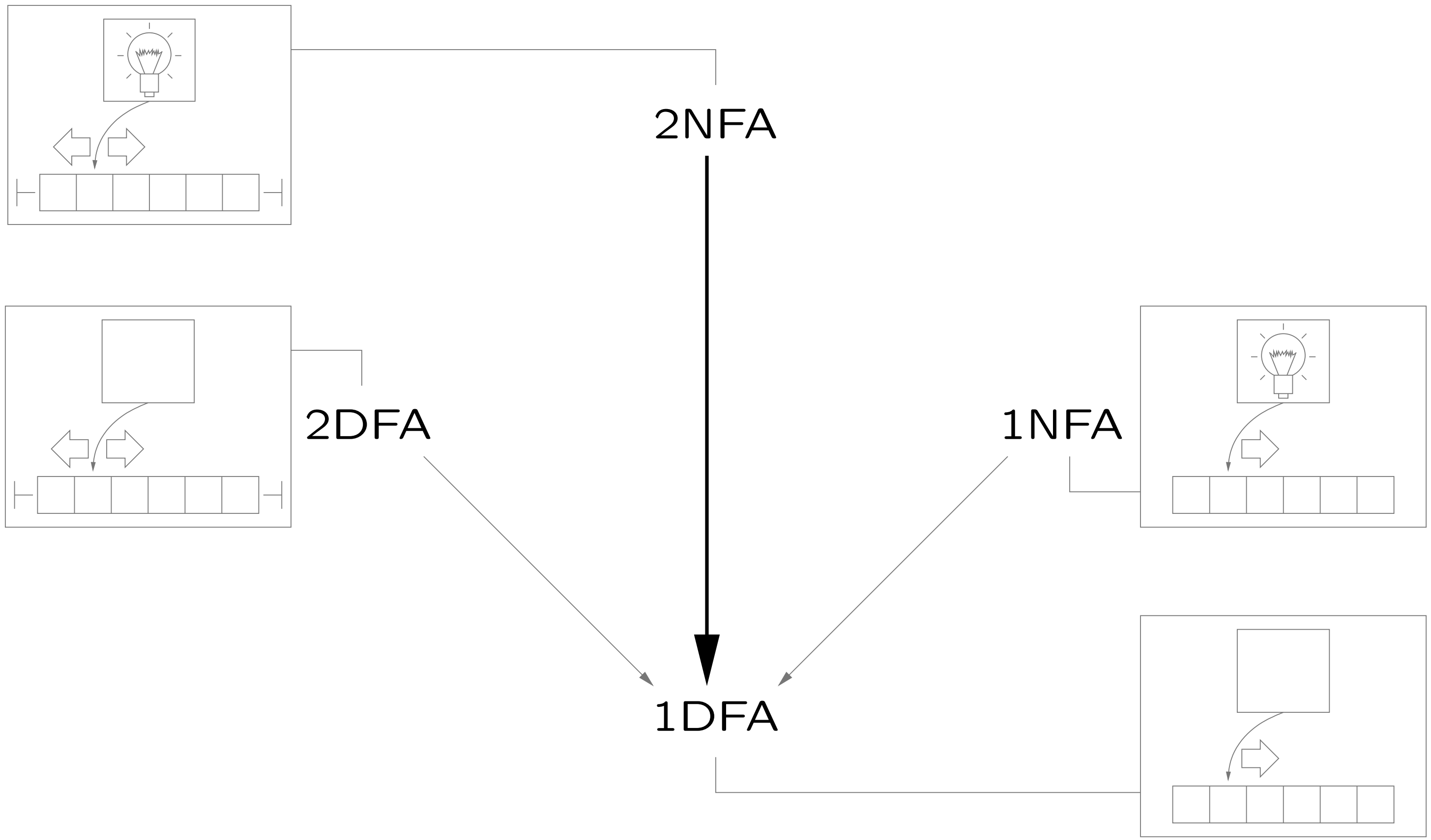
$2^n - 1$

the big picture

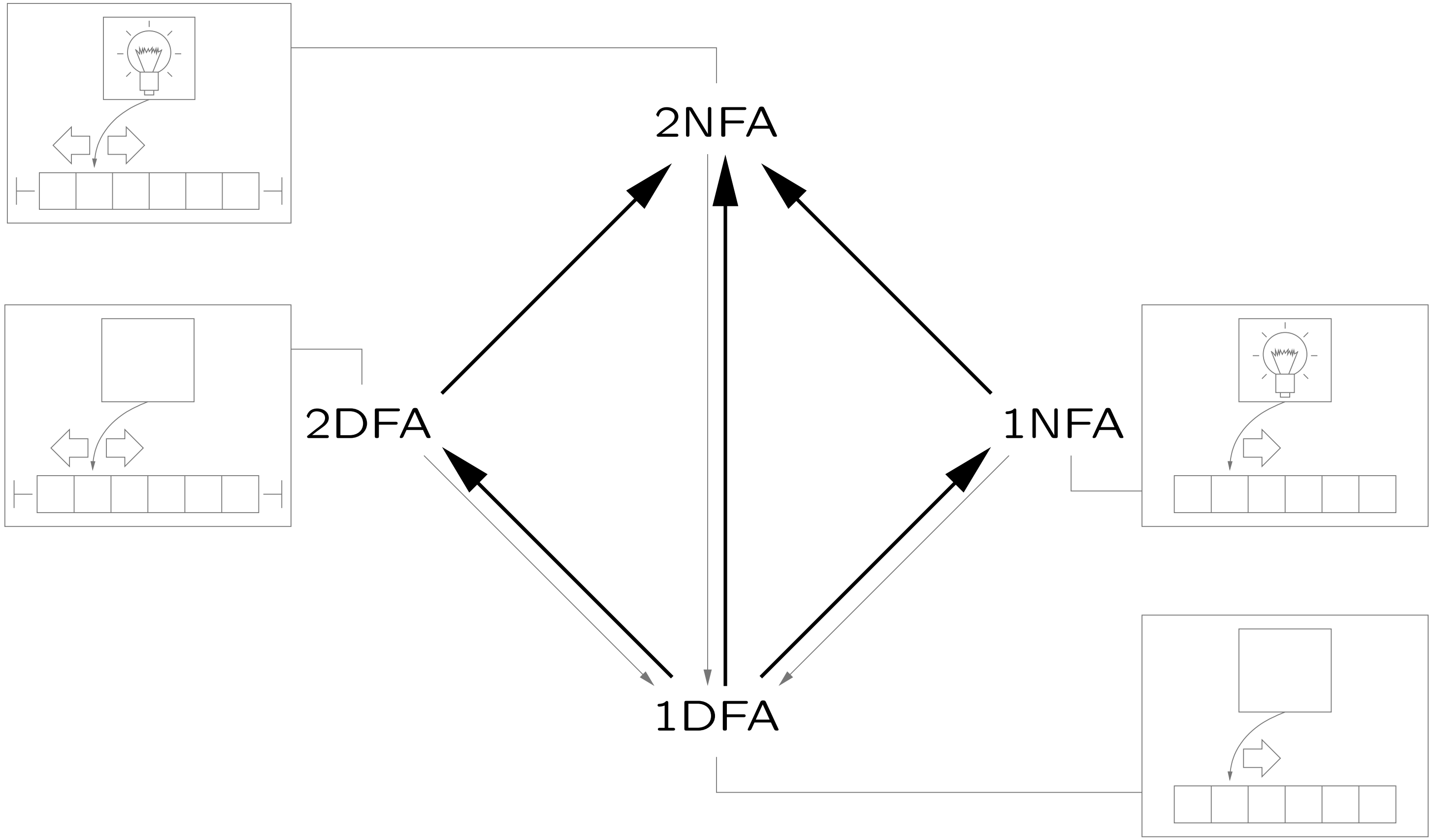


$2^n - 1$, $n(n^n - (n-1)^n)$

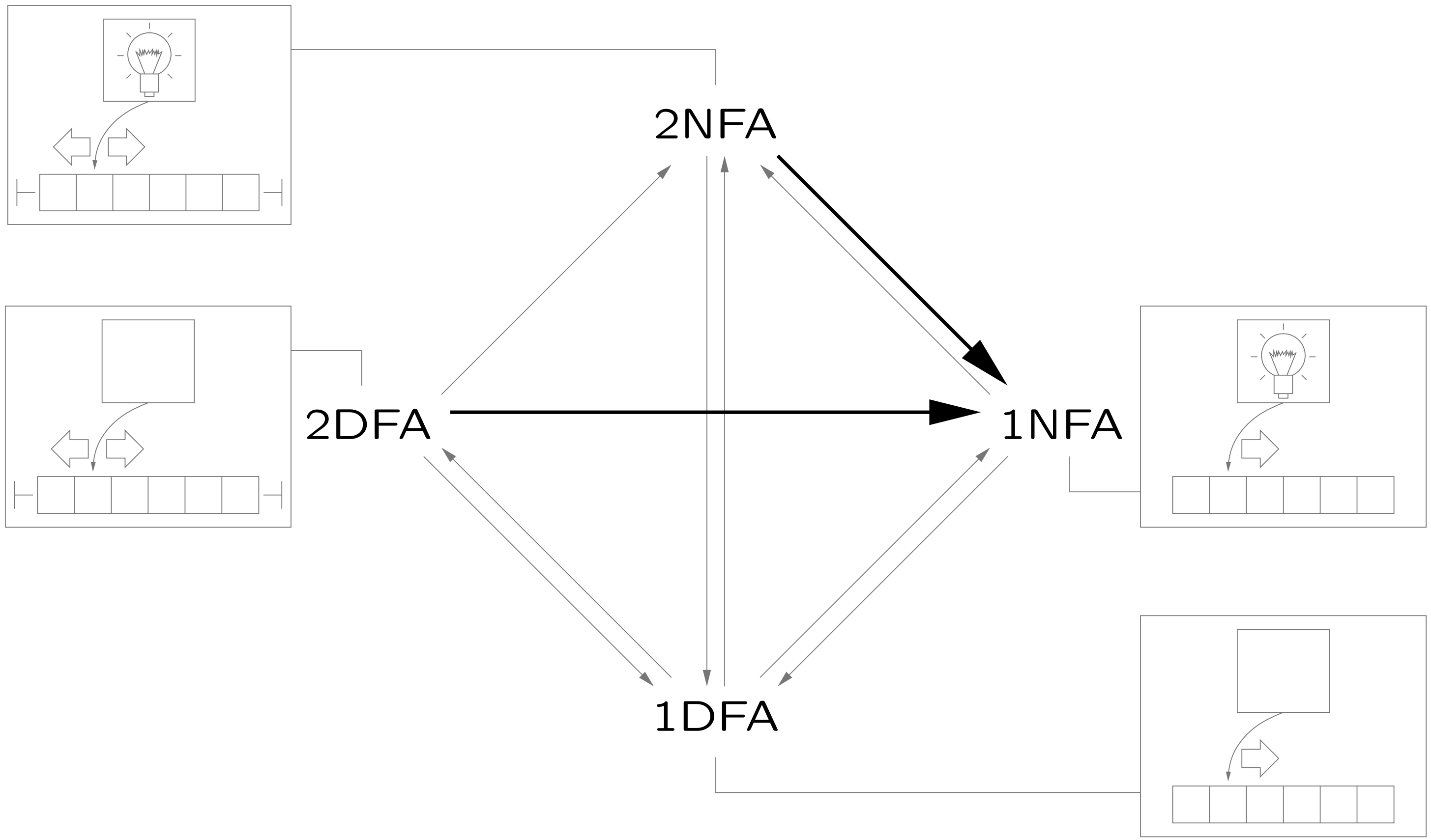
the big picture



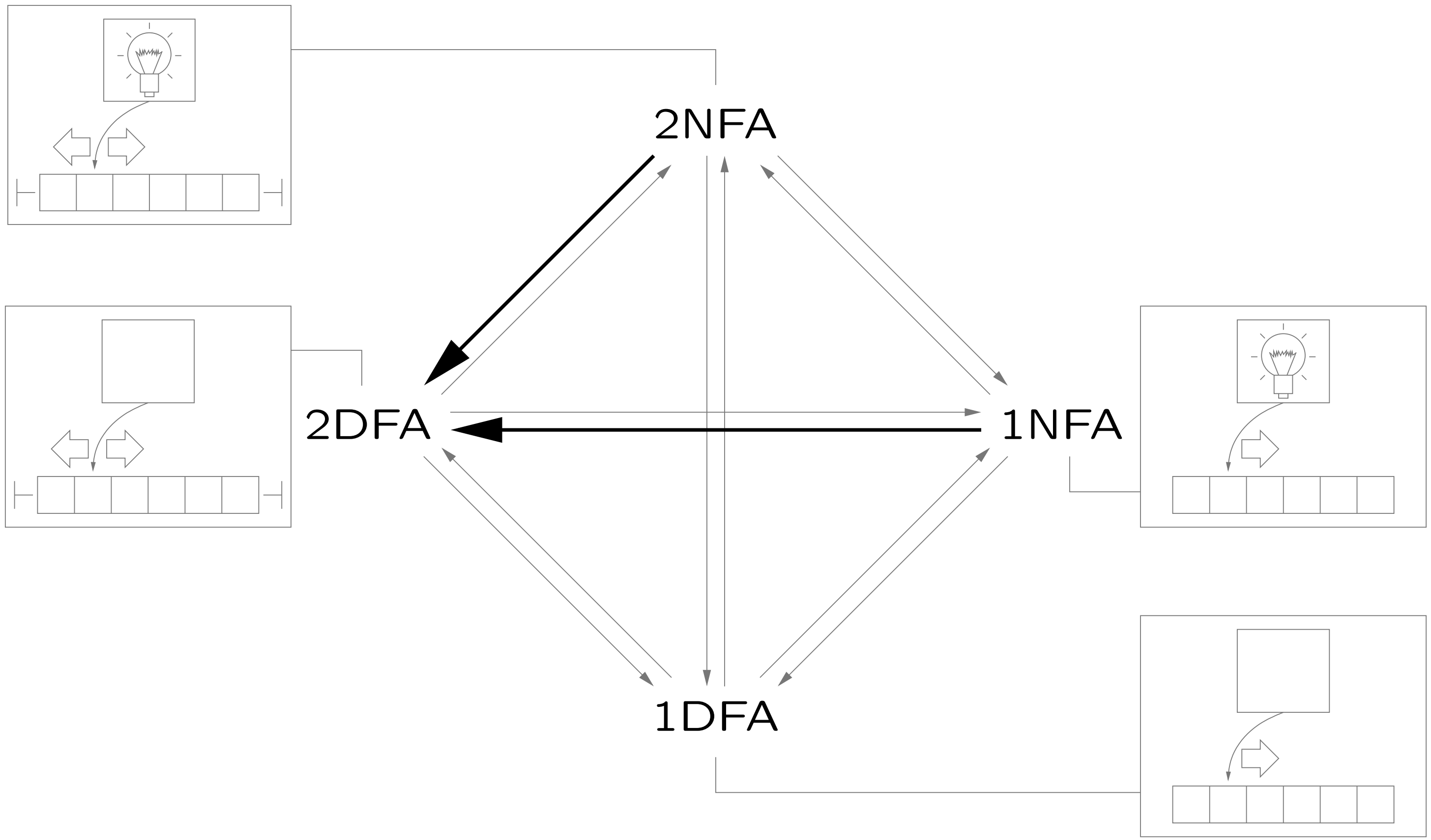
$2^n - 1,\ n(n^n - (n-1)^n),\ \sum_{i<n}\sum_{j<n}\binom{n}{i}\binom{n}{j}(2^i - 1)^j$



$2^n - 1$, $n(n^n - (n-1)^n)$, $\sum_{i<n} \sum_{j<n} \binom{n}{i}\binom{n}{j}(2^i - 1)^j$, n

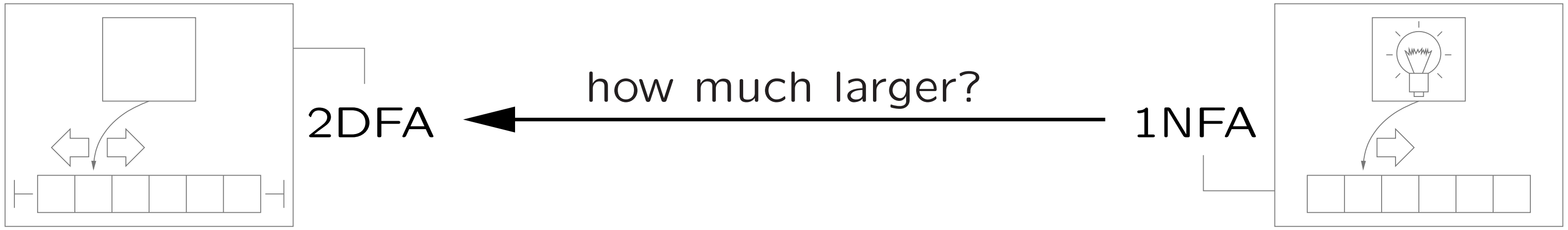


$2^n - 1$, $n(n^n - (n-1)^n)$, $\sum_{i<n} \sum_{j<n} \binom{n}{i}\binom{n}{j}(2^i - 1)^j$, n, $\binom{2n}{n+1}$

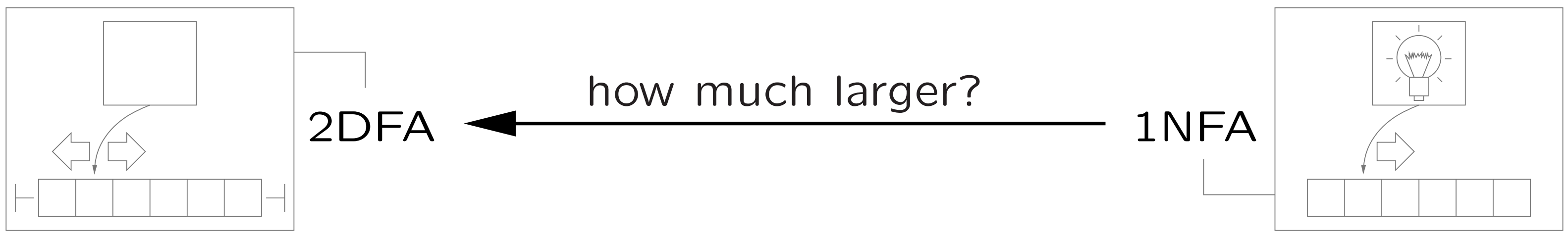


$2^n - 1$, $n(n^n - (n-1)^n)$, $\sum_{i<n}\sum_{j<n}\binom{n}{i}\binom{n}{j}(2^i-1)^j$, n, $\binom{2n}{n+1}$, ?, ?

the problem

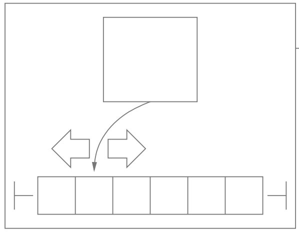


the problem

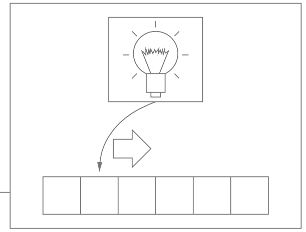


polynomially ? exponentially

the problem



2DFA ← how much larger? — 1NFA



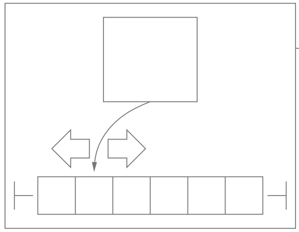
polynomially ? exponentially

determ Turing machine

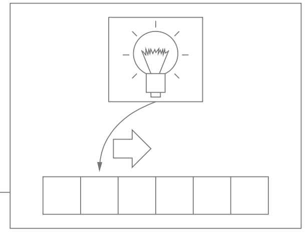
DTM ← how much slower? — NTM

nondeterm Turing machine

the problem



2DFA $\xleftarrow{\text{how much larger?}}$ 1NFA



polynomially ? exponentially

---

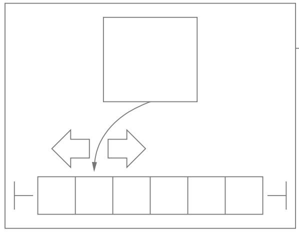
determ Turing machine

DTM $\xleftarrow{\text{how much slower?}}$ NTM

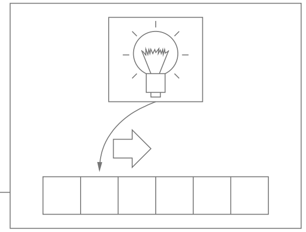
nondeterm Turing machine

polynomially ? exponentially

the problem



2DFA ← how much larger? — 1NFA



polynomially ? exponentially

---

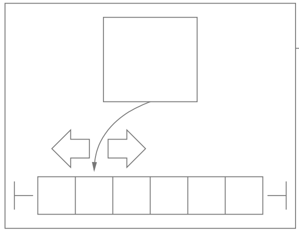
determ Turing machine

DTM ← how much slower? — NTM

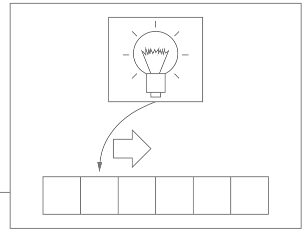
nondeterm Turing machine

polynomially ? exponentially

P=NP P≠NP



2DFA ← how much larger? ← 1NFA



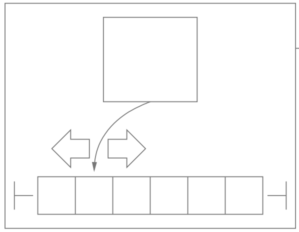
polynomially ? exponentially

determ Turing machine — DTM ← how much slower? ← NTM — nondeterm Turing machine

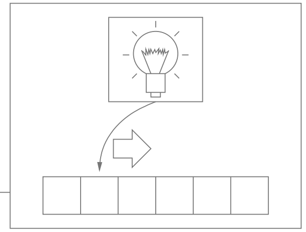
SATISFIABILITY

$$\overset{n}{\overleftrightarrow{(x_1 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_1) \wedge (x_3 \vee x_1 \vee x_2)}}$$

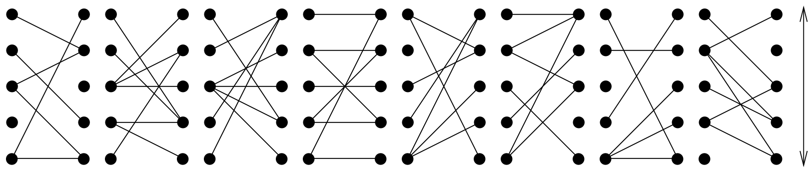
is there a satisfying assignment?



2DFA ← how much larger? — 1NFA



LIVENESS



n

is there a live path?

---

determ Turing machine

DTM ← how much slower? — NTM

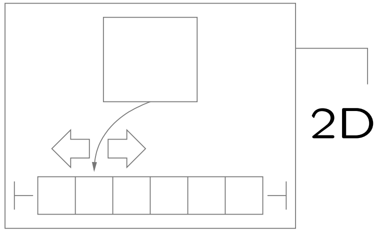
nondeterm Turing machine

SATISFIABILITY

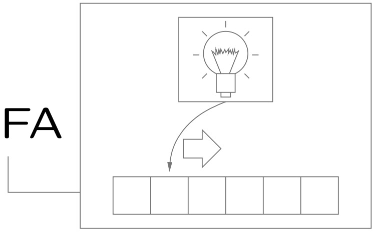
$$\overset{n}{\longleftrightarrow}$$

$$(x_1 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_1) \wedge (x_3 \vee x_1 \vee x_2)$$

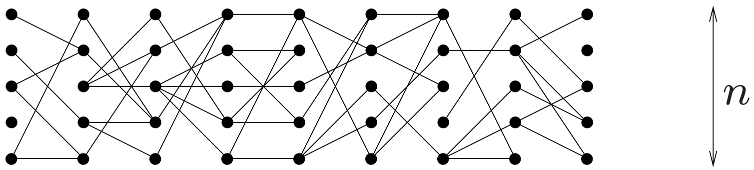
is there a satisfying assignment?



2DFA ← how much larger? — 1NFA



LIVENESS



is there a live path?

---

determ Turing machine

DTM ← how much slower? — NTM

nondeterm Turing machine

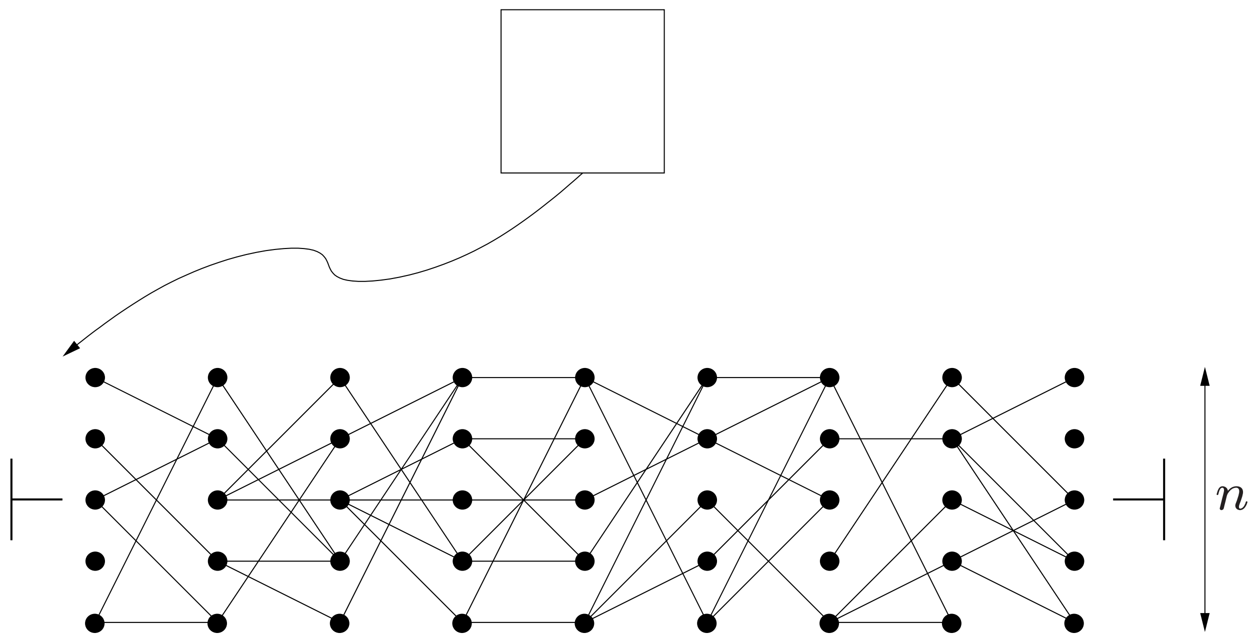
SATISFIABILITY

$$\overset{n}{\longleftrightarrow}$$

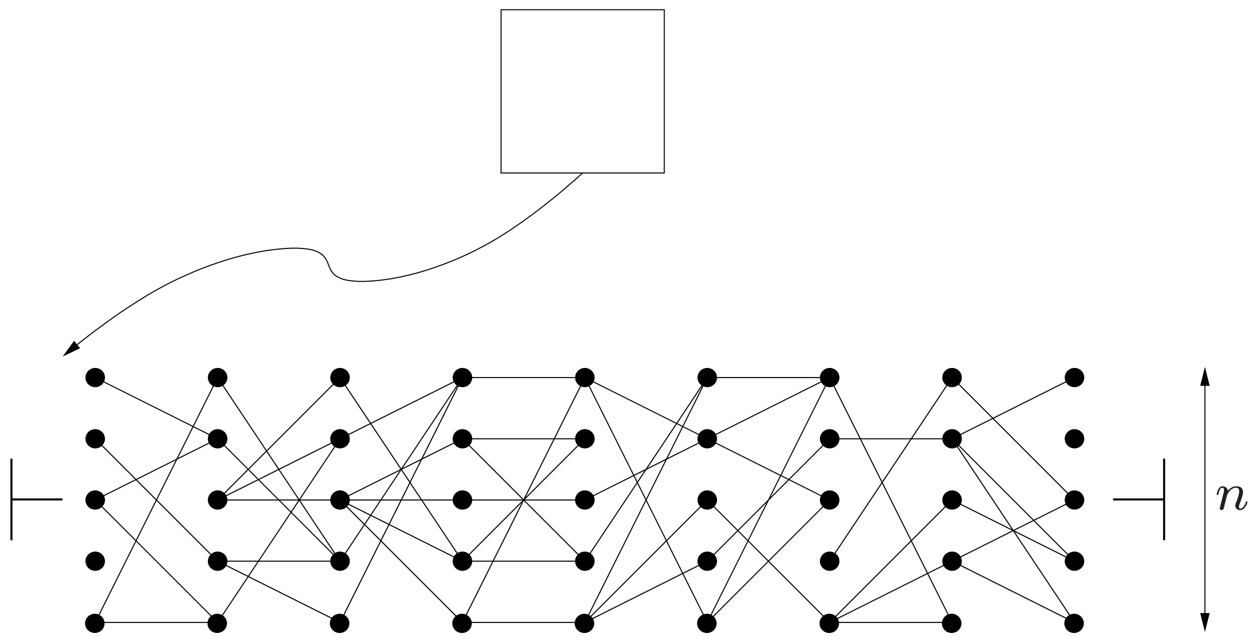
$$(x_1 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_1) \wedge (x_3 \vee x_1 \vee x_2)$$

is there a satisfying assignment?

2DFA against LIVENESS

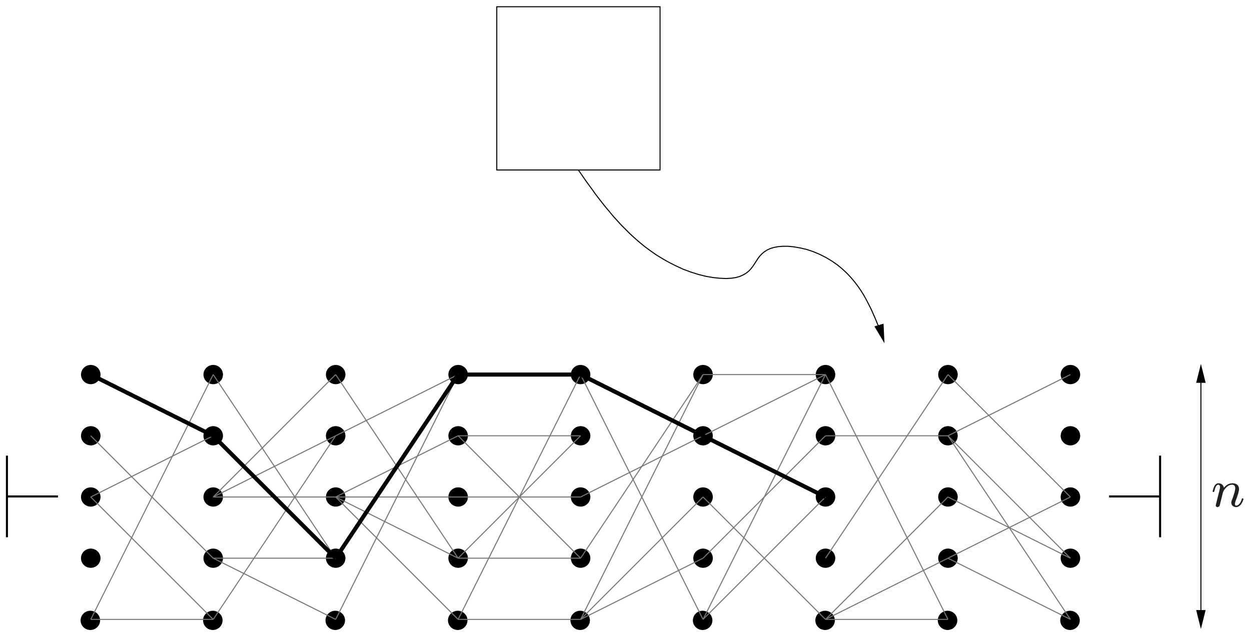


2DFA against LIVENESS: graph exploration



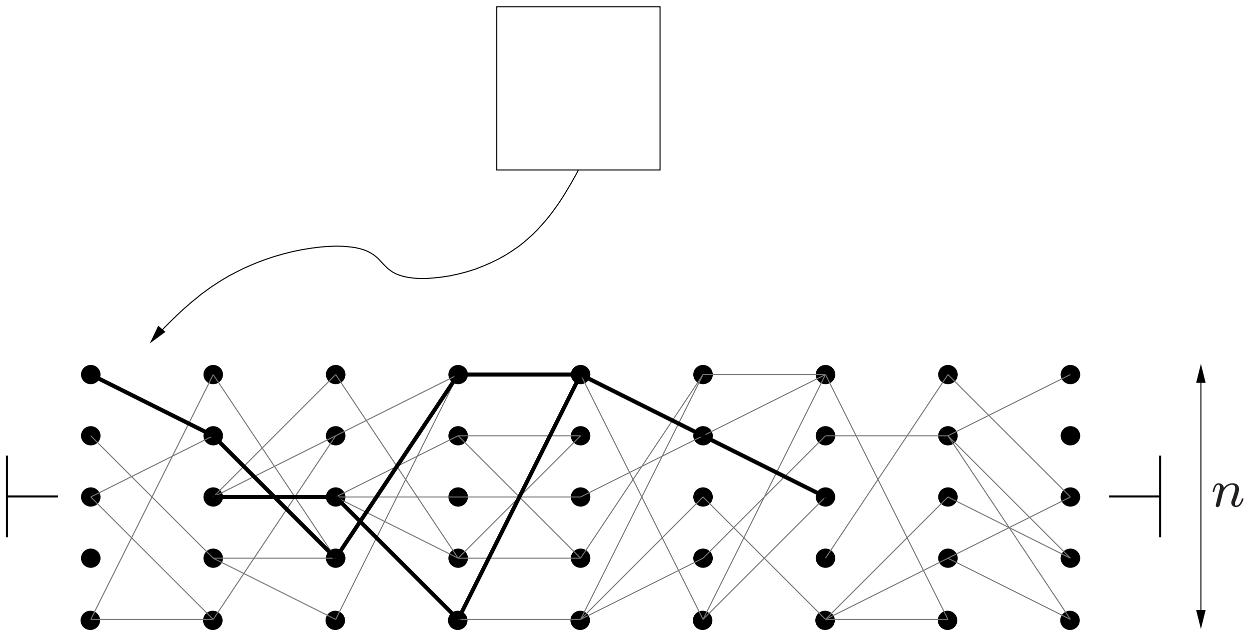
IDEA: depth first search!

2DFA against LIVENESS: graph exploration



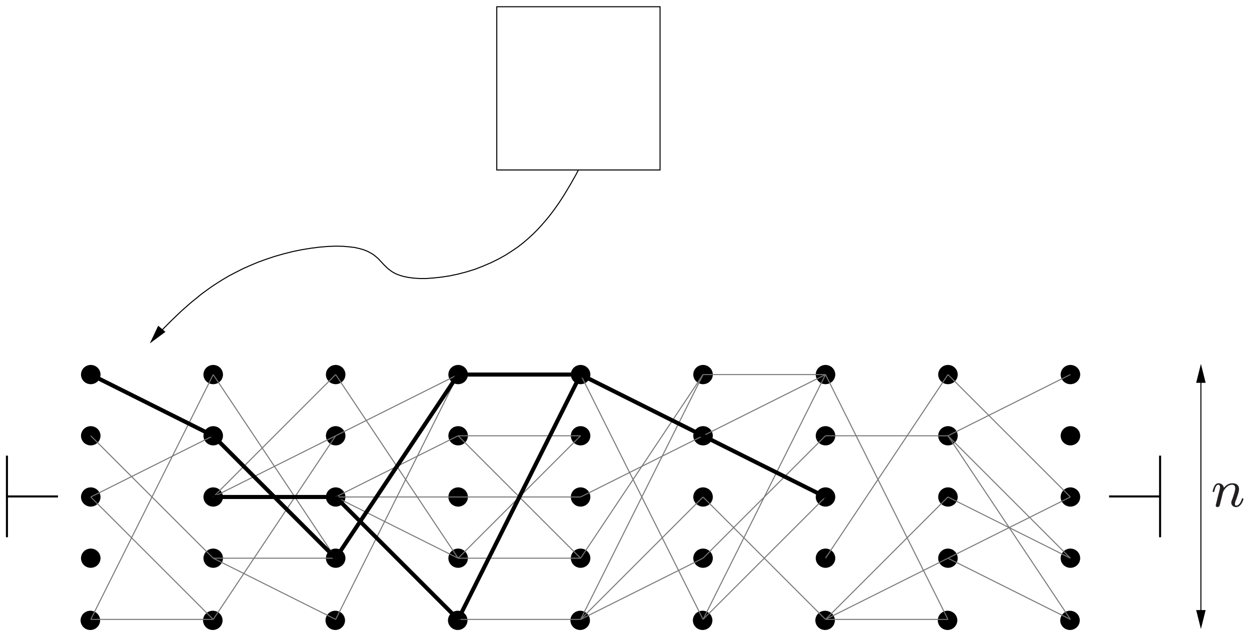
IDEA: depth first search!

2DFA against LIVENESS: graph exploration



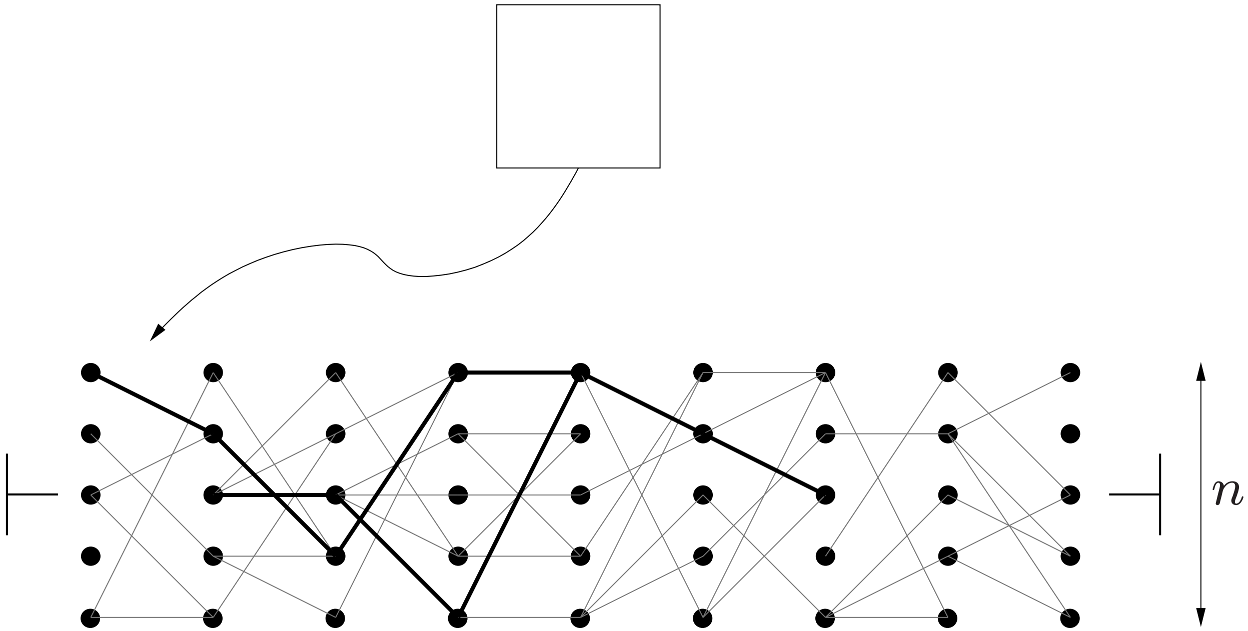
IDEA: depth first search!

2DFA against LIVENESS: graph exploration



IDEA: depth first search!
PROBLEM: we get lost

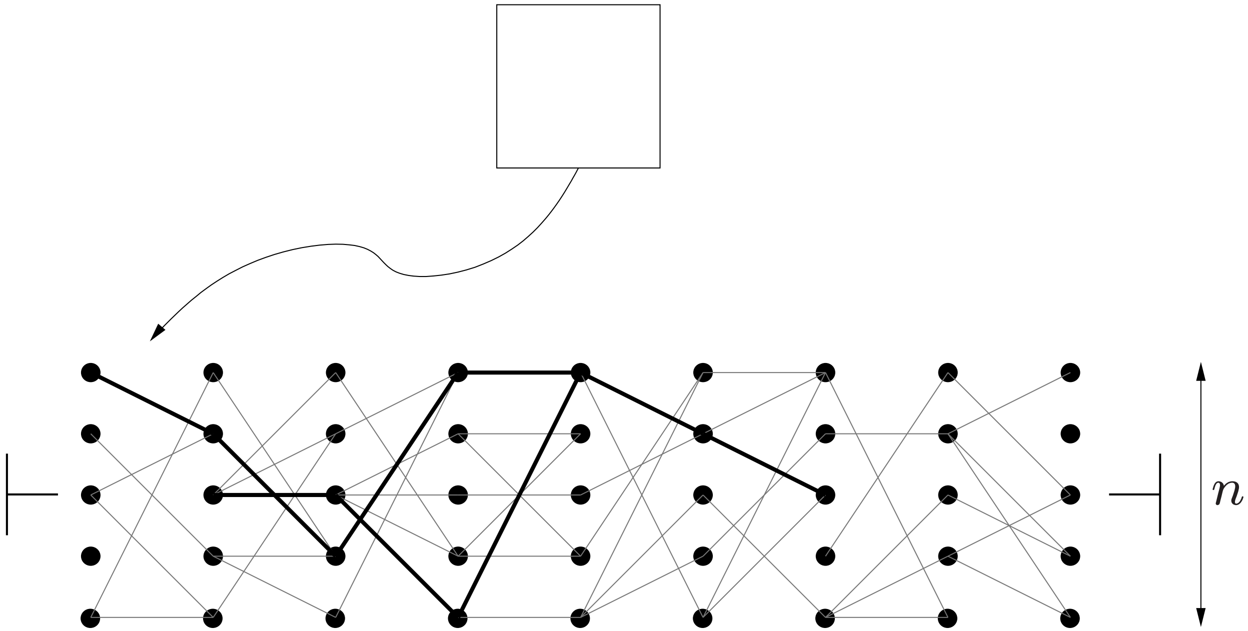
2DFA against LIVENESS: graph exploration



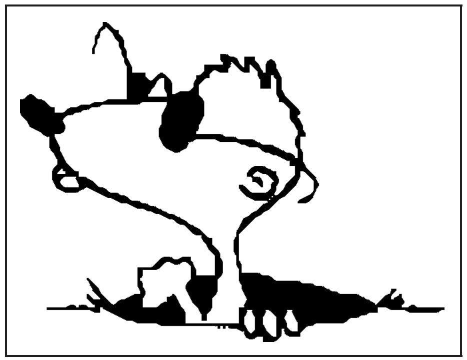
IDEA: depth first search!
PROBLEM: we get lost

THEOREM: no graph exploration can work

2DFA against LIVENESS: graph exploration

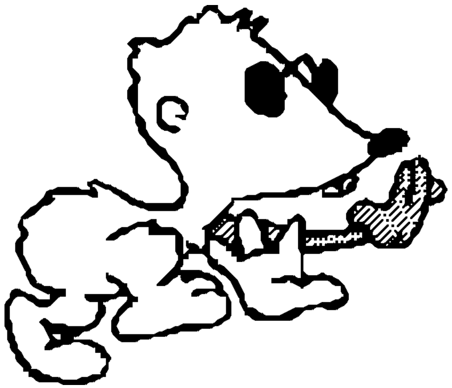


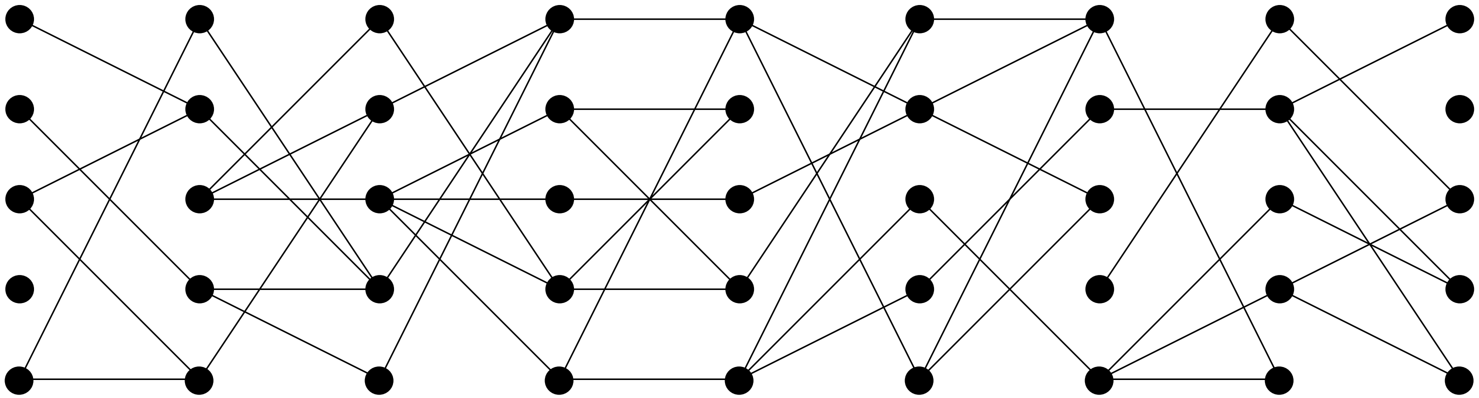
IDEA: depth first search!
PROBLEM: we get lost



THEOREM: no graph exploration can work

MOLE against LIVENESS





PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits

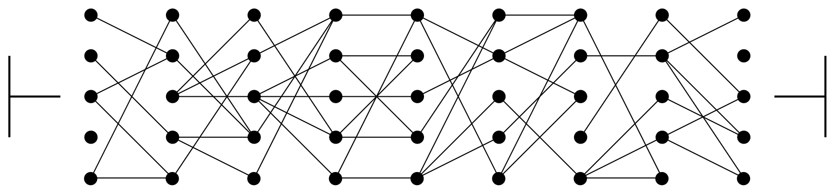
PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits

w
(dead)

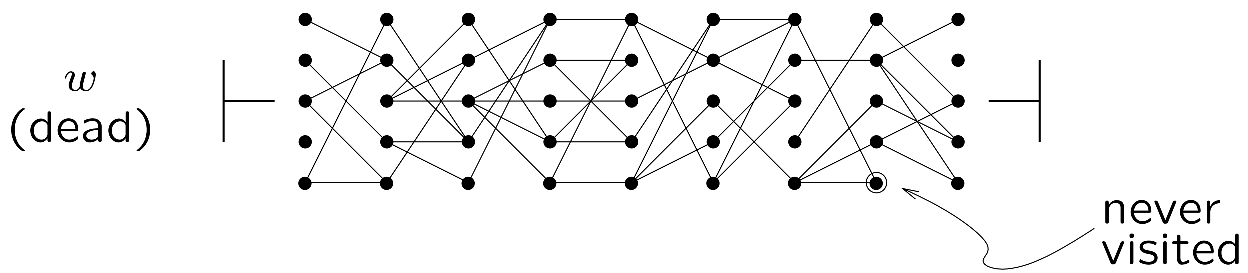


PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits

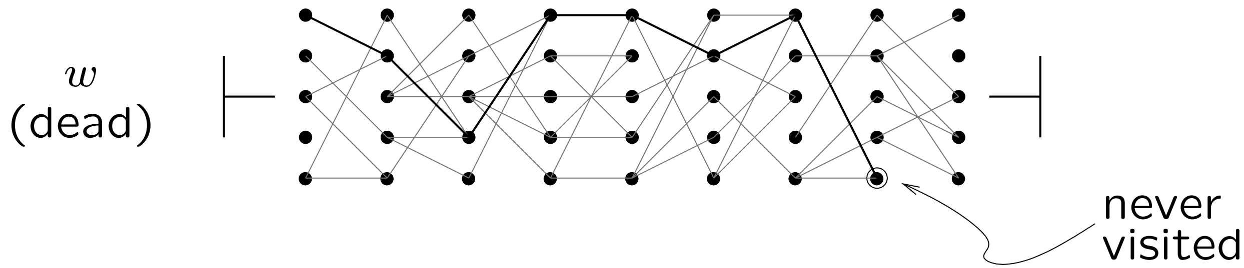


PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\Longleftrightarrow$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits

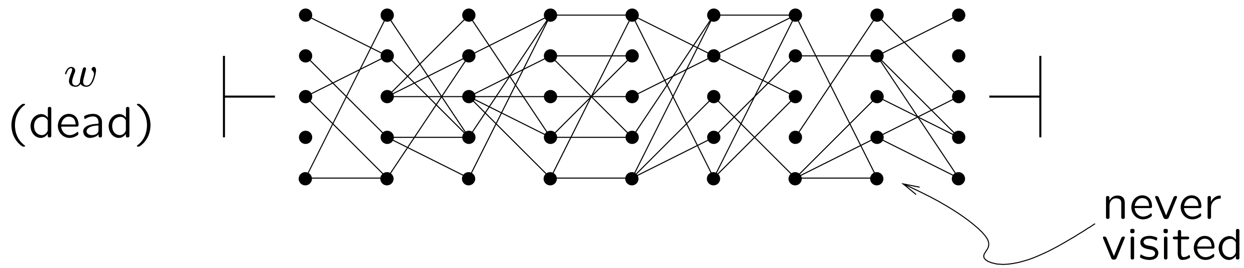


PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits



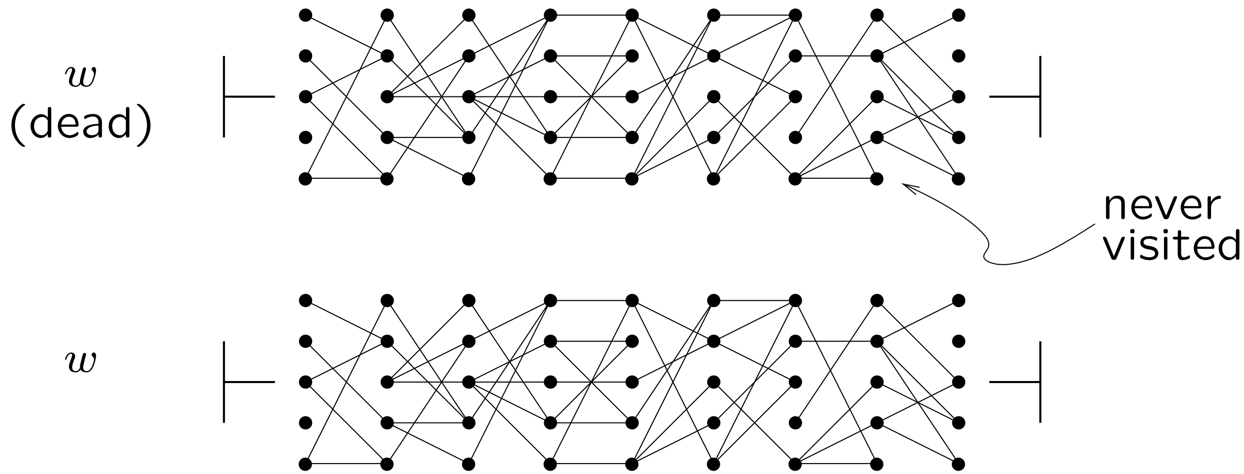
MOLE against LIVENESS

PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits

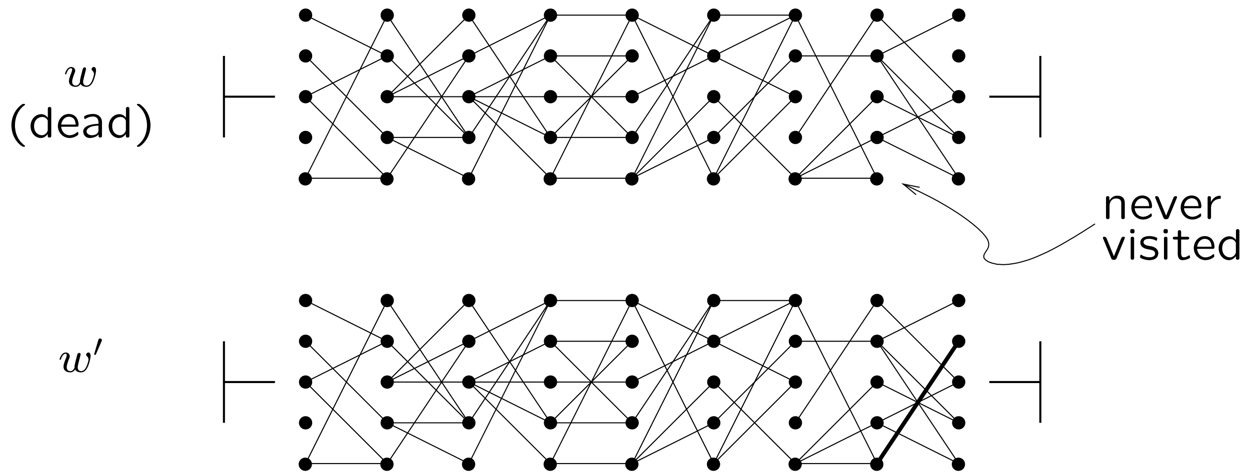


PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits



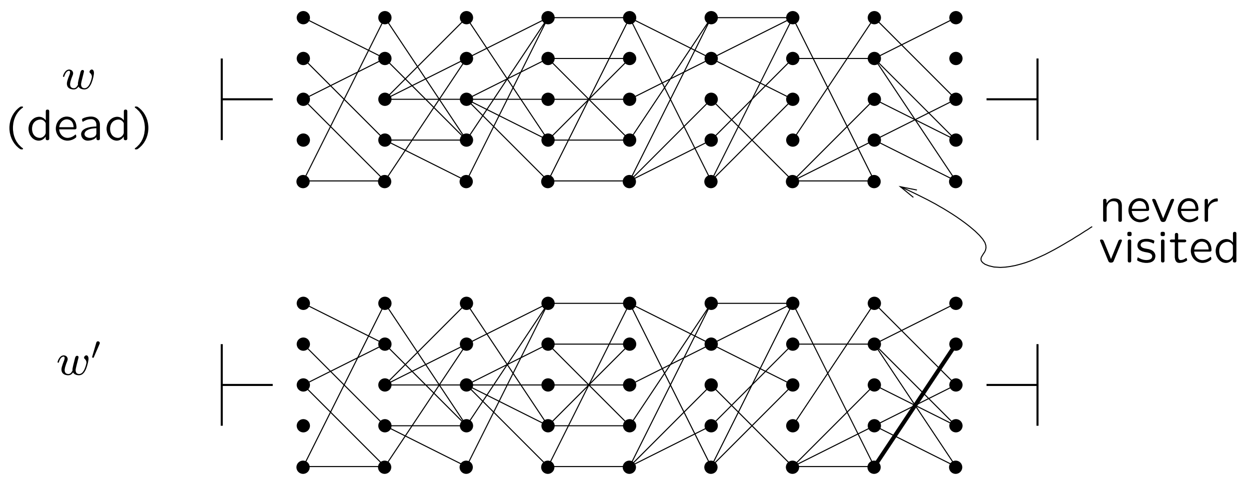
MOLE against LIVENESS

PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits



same computation!

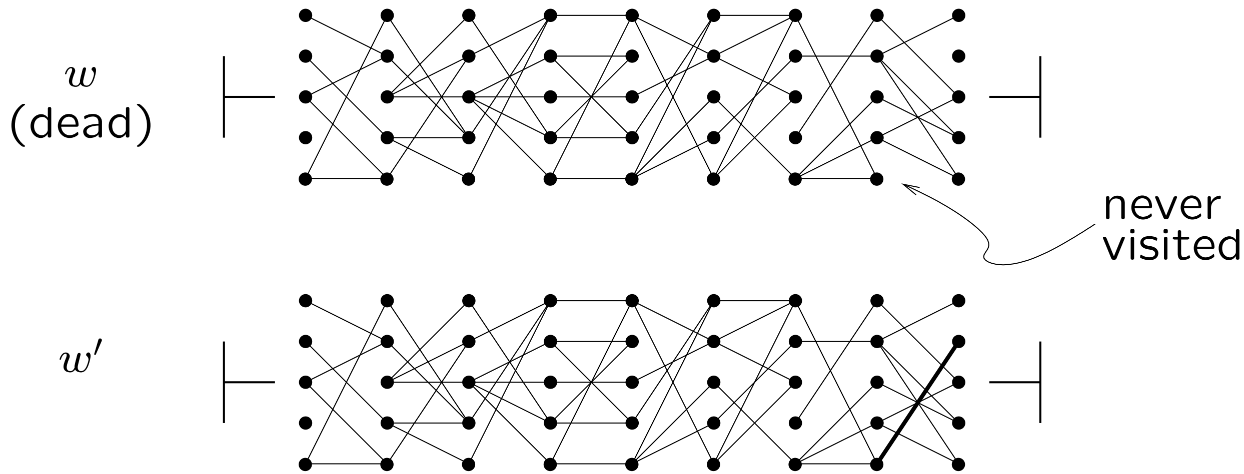
MOLE against LIVENESS

PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits



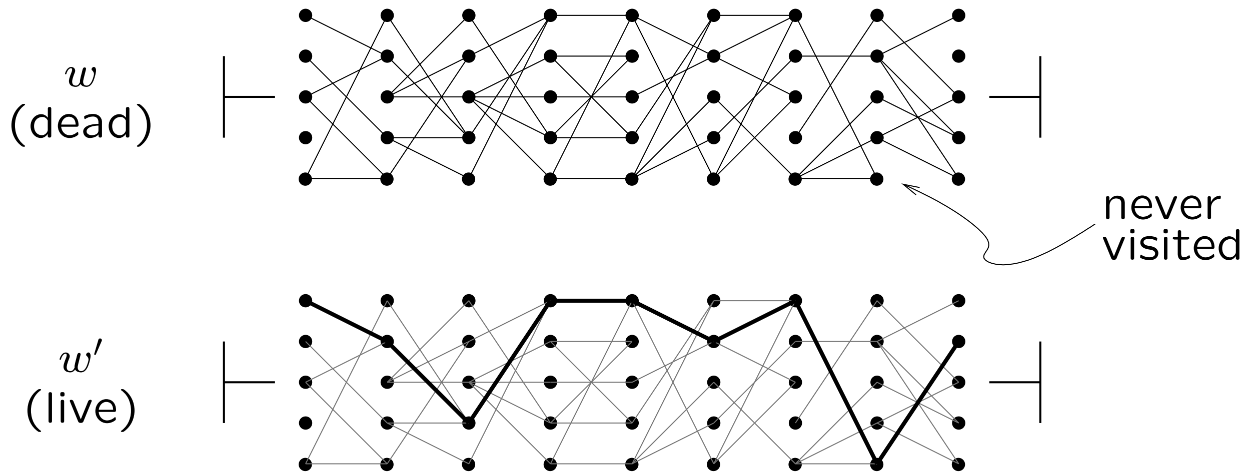
same computation $\Rightarrow$ same decision

PROOF PLAN: Construct a string of graphs w such that:

the mole fails on w

$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

$\Longleftarrow$ w is dead and contains a live node that the mole never visits



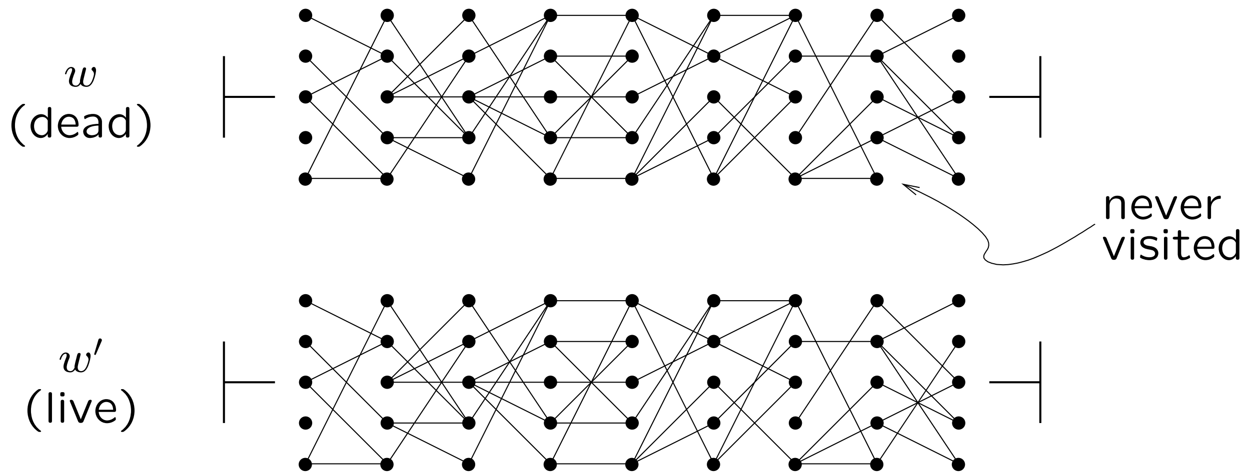
same computation $\Rightarrow$ same decision

PROOF PLAN: Construct a string of graphs w such that:

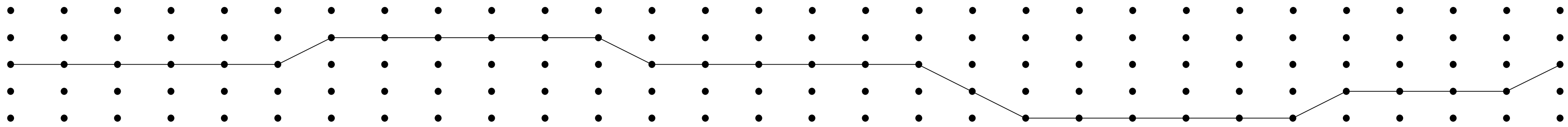
the mole fails on w

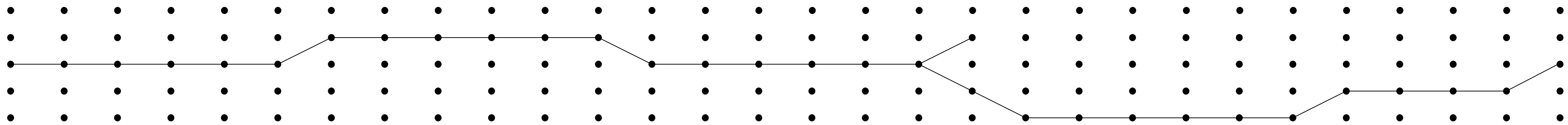
$\iff$ w is live and the mole rejects $\vee$ w is dead and the mole accepts

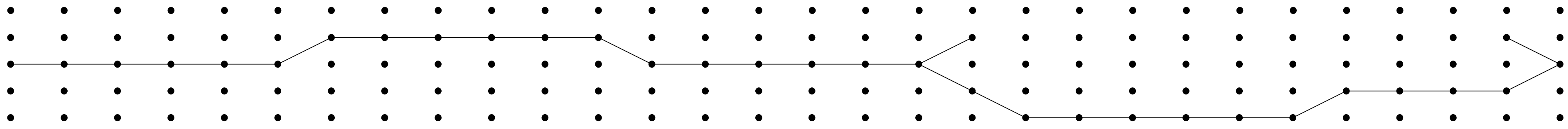
$\Longleftarrow$ w is dead and contains a live node that the mole never visits



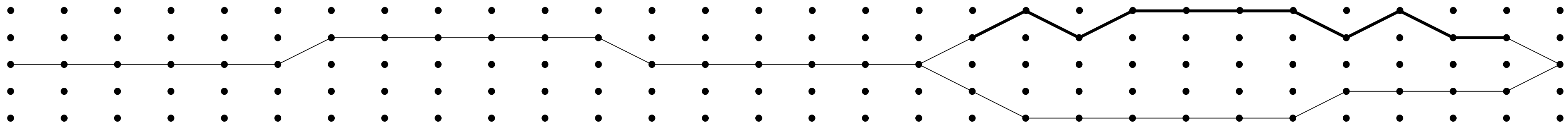
same computation $\Rightarrow$ same decision

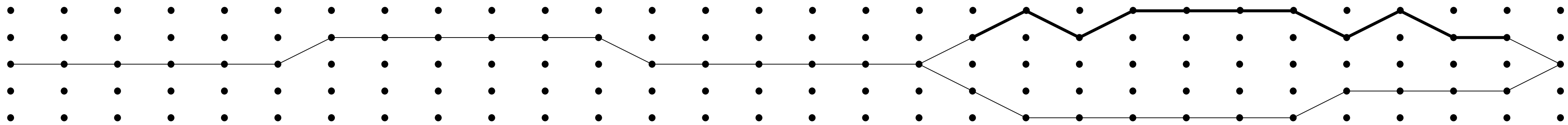


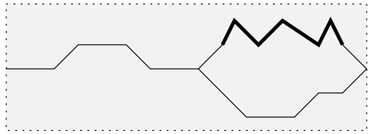


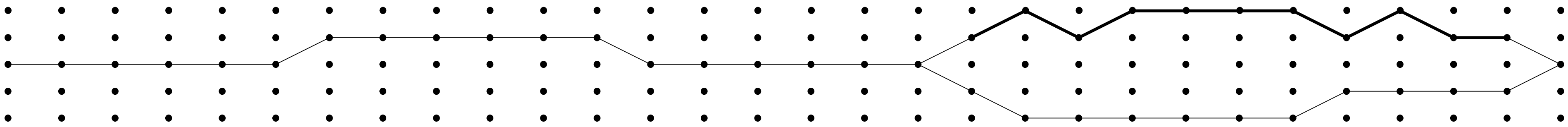


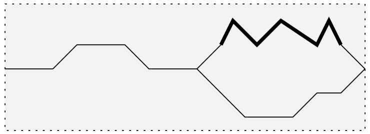
the main argument

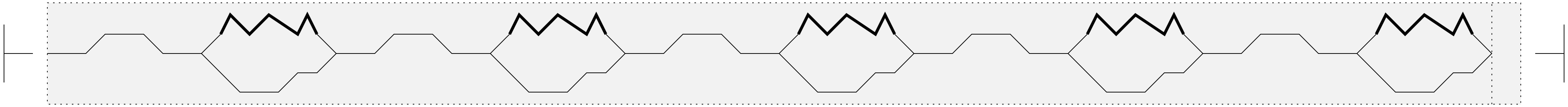


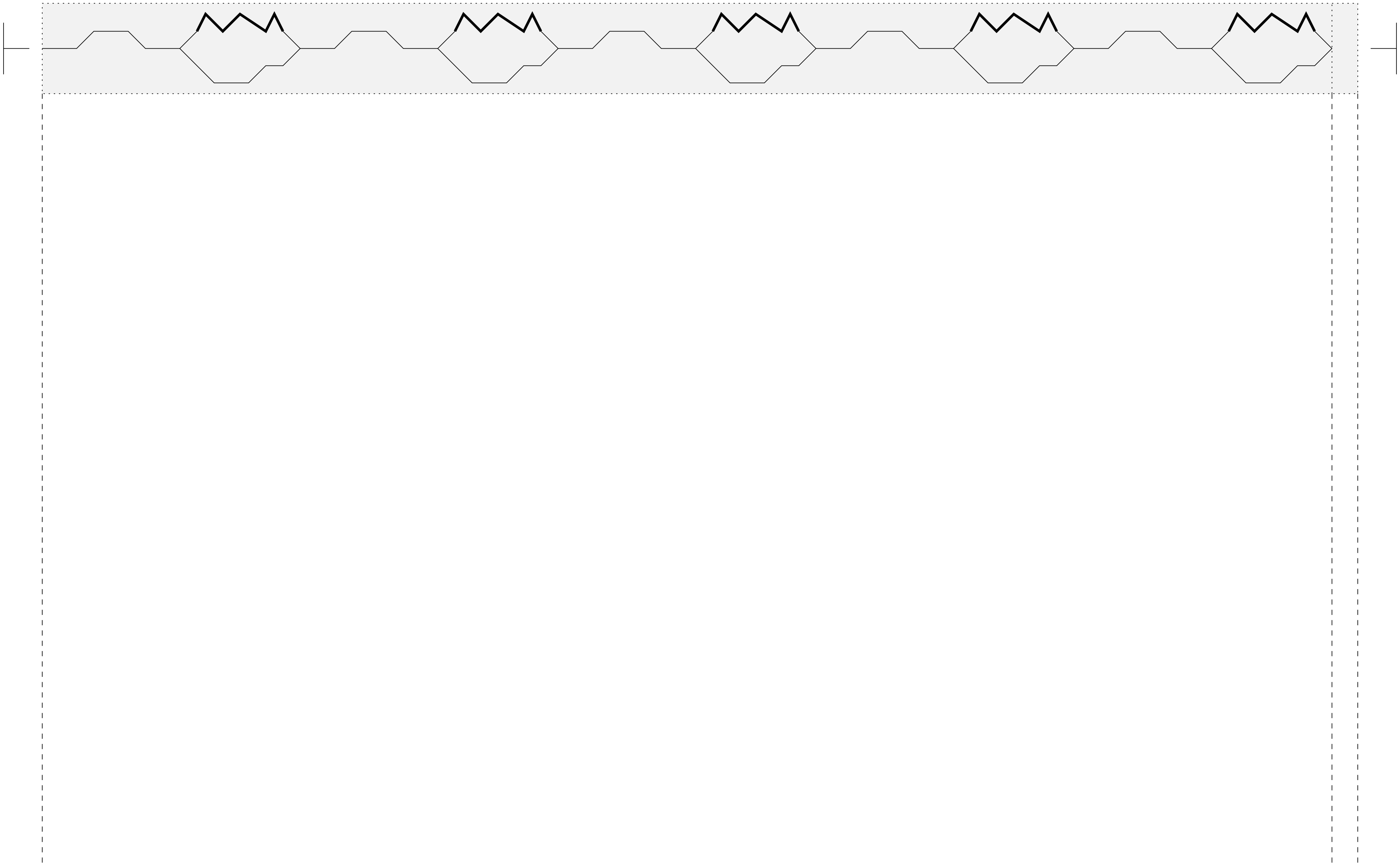


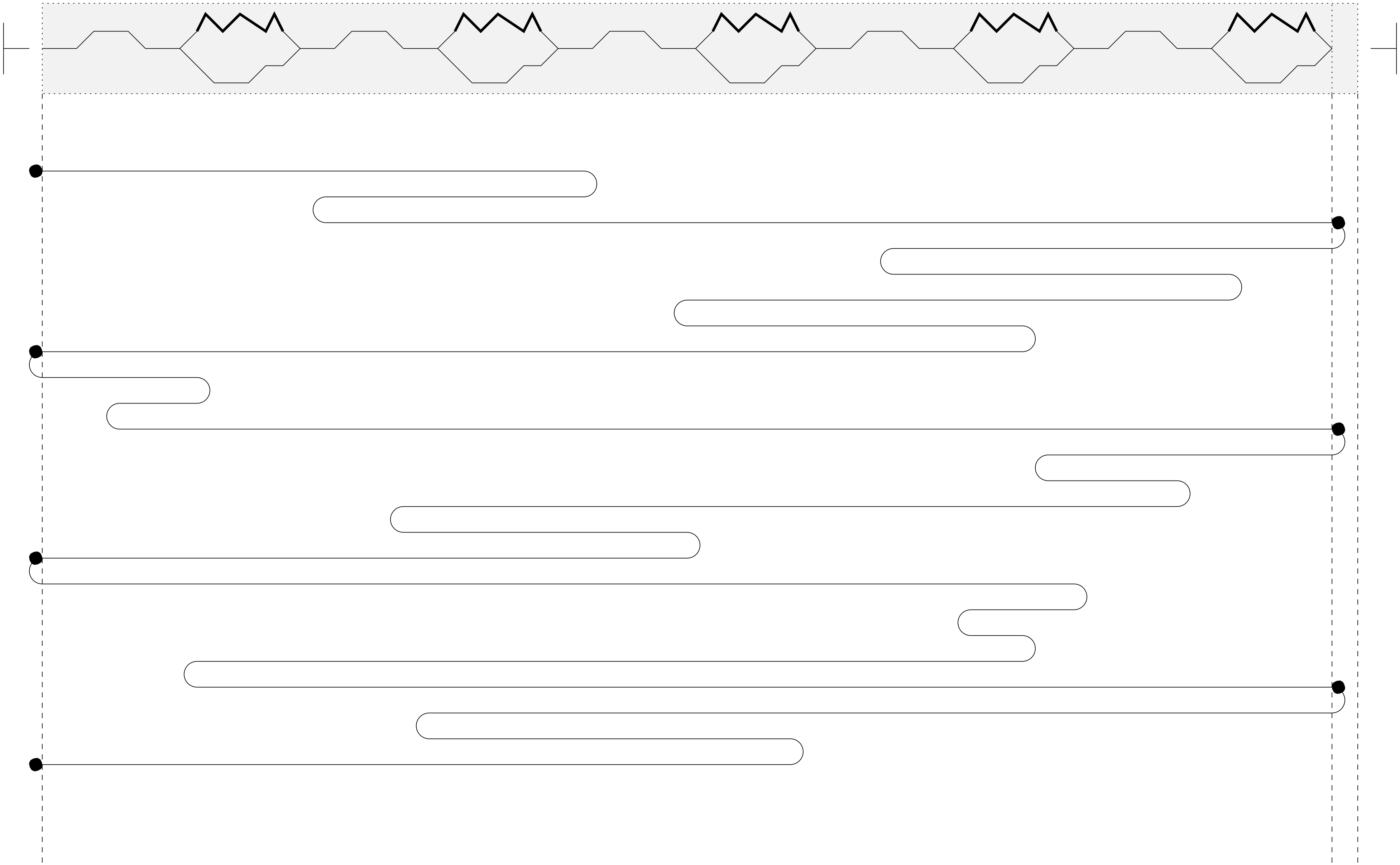




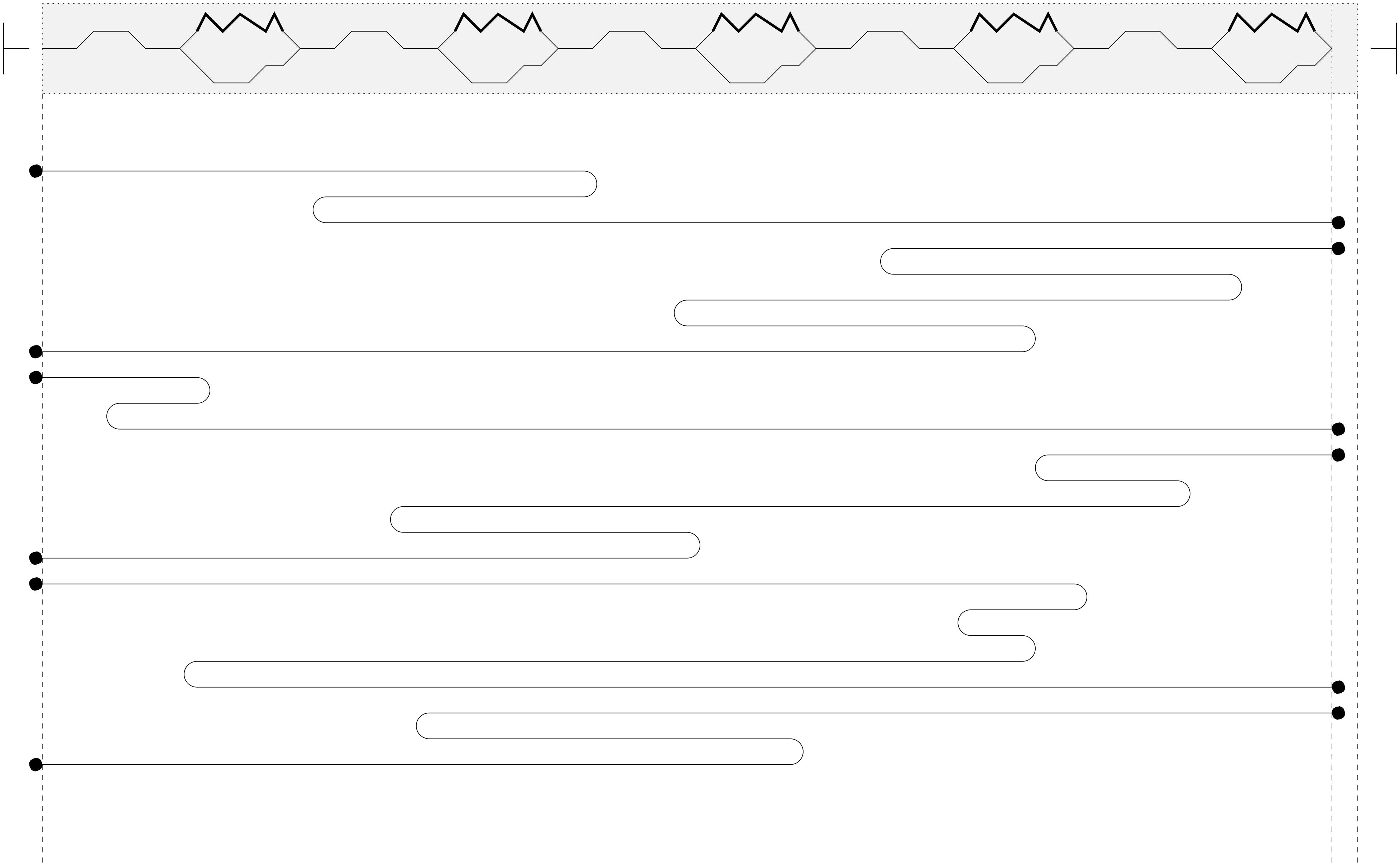








the main argument



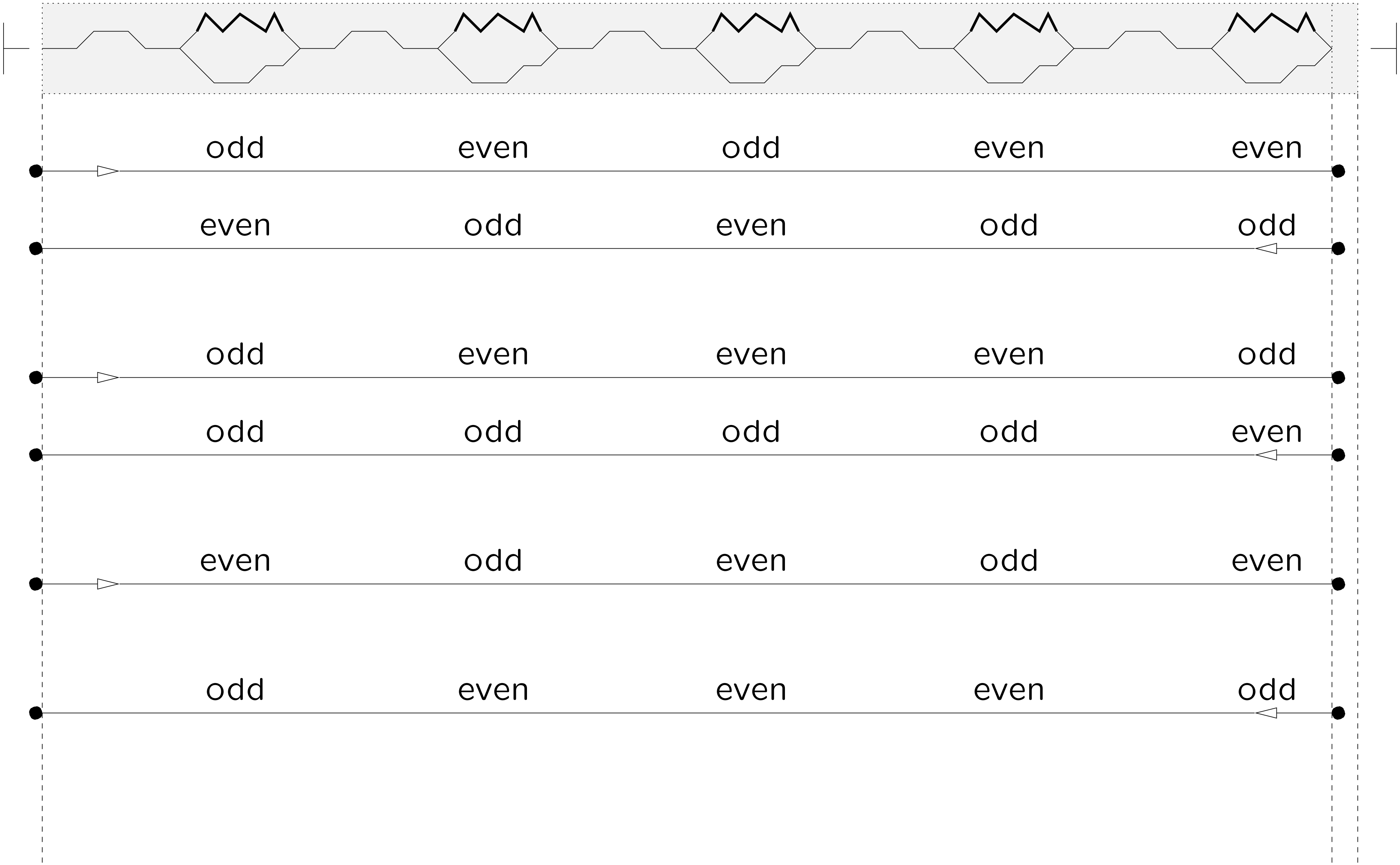
the main argument


odd

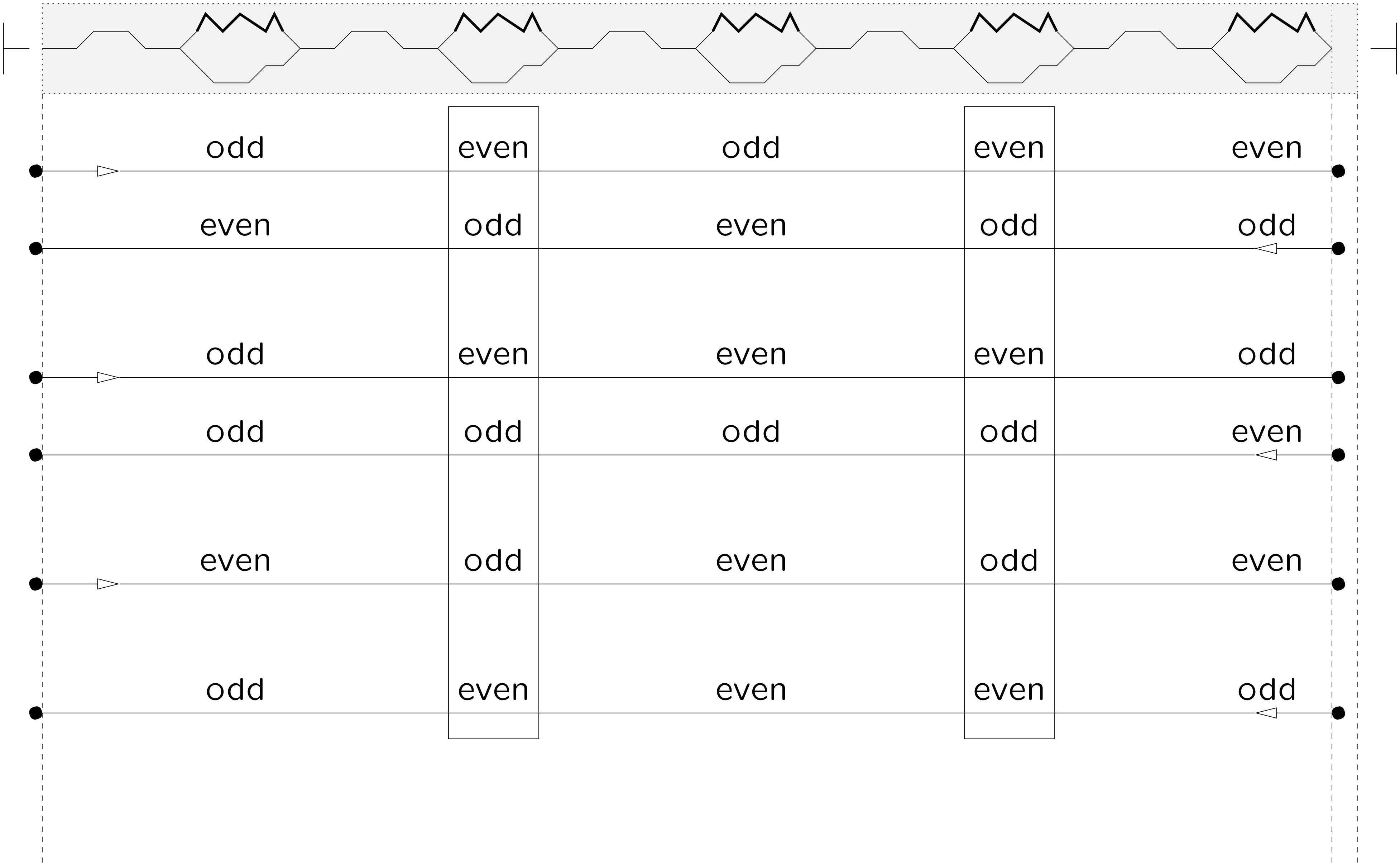
the main argument

the main argument

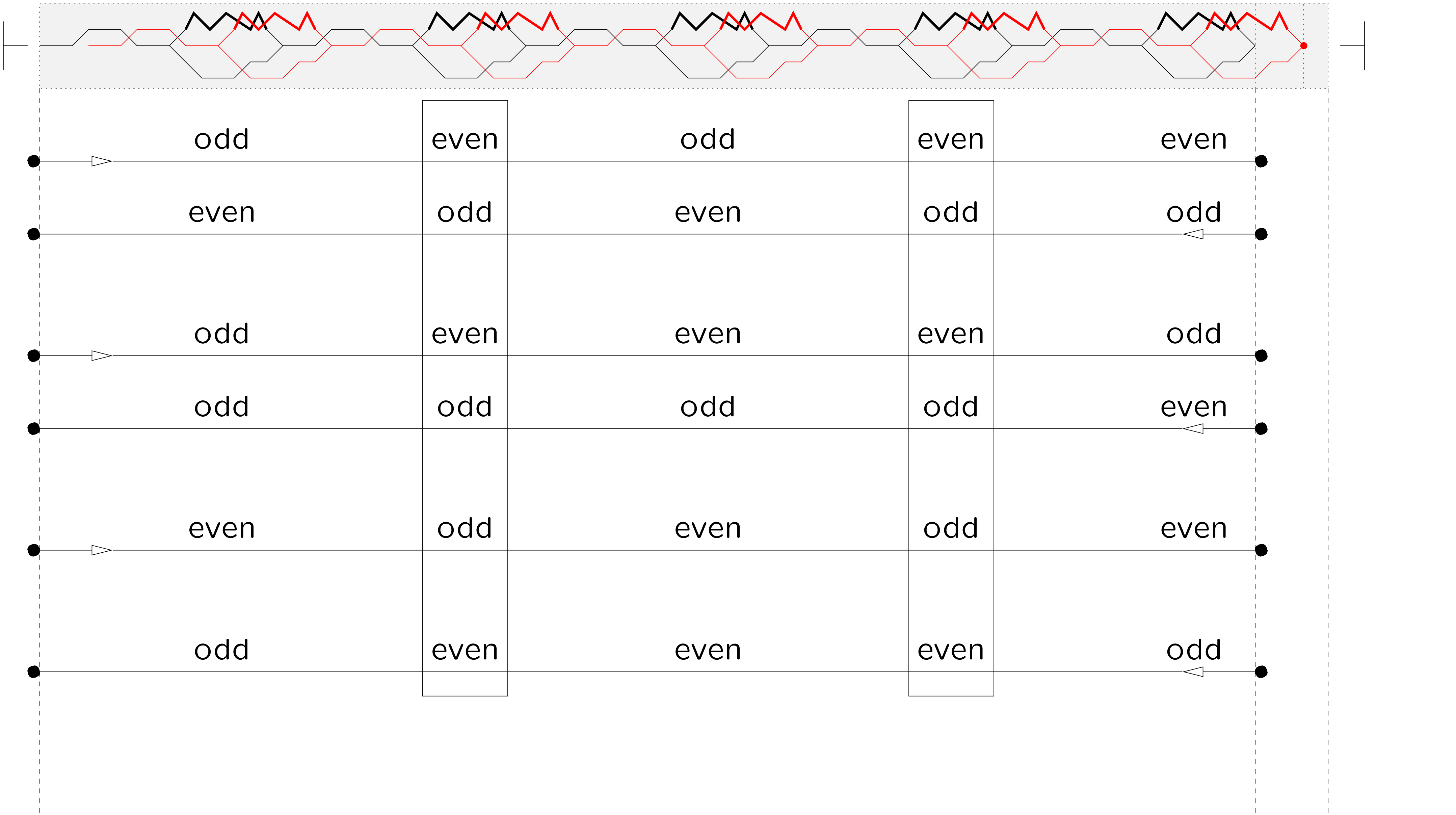
the main argument

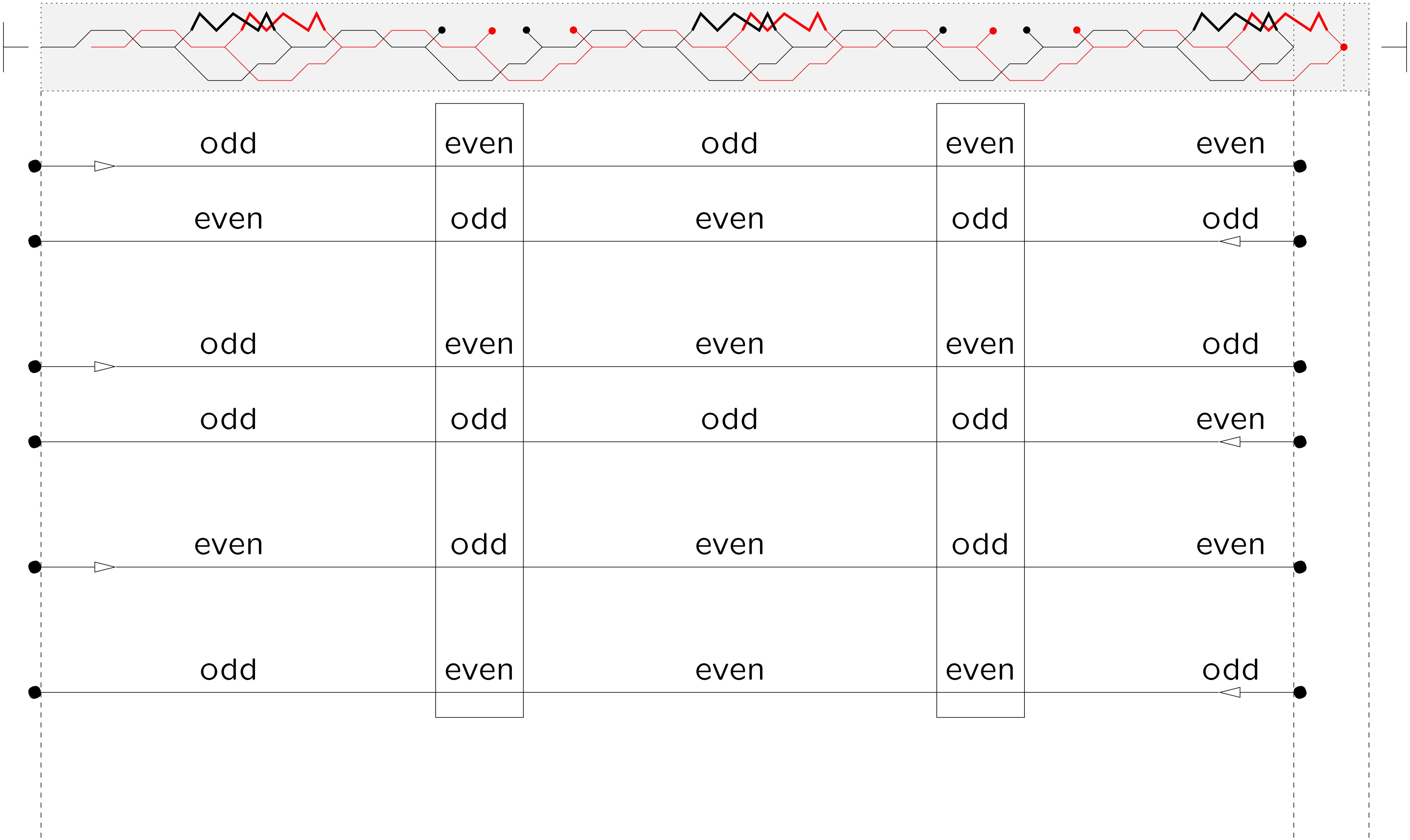


the main argument



the main argument



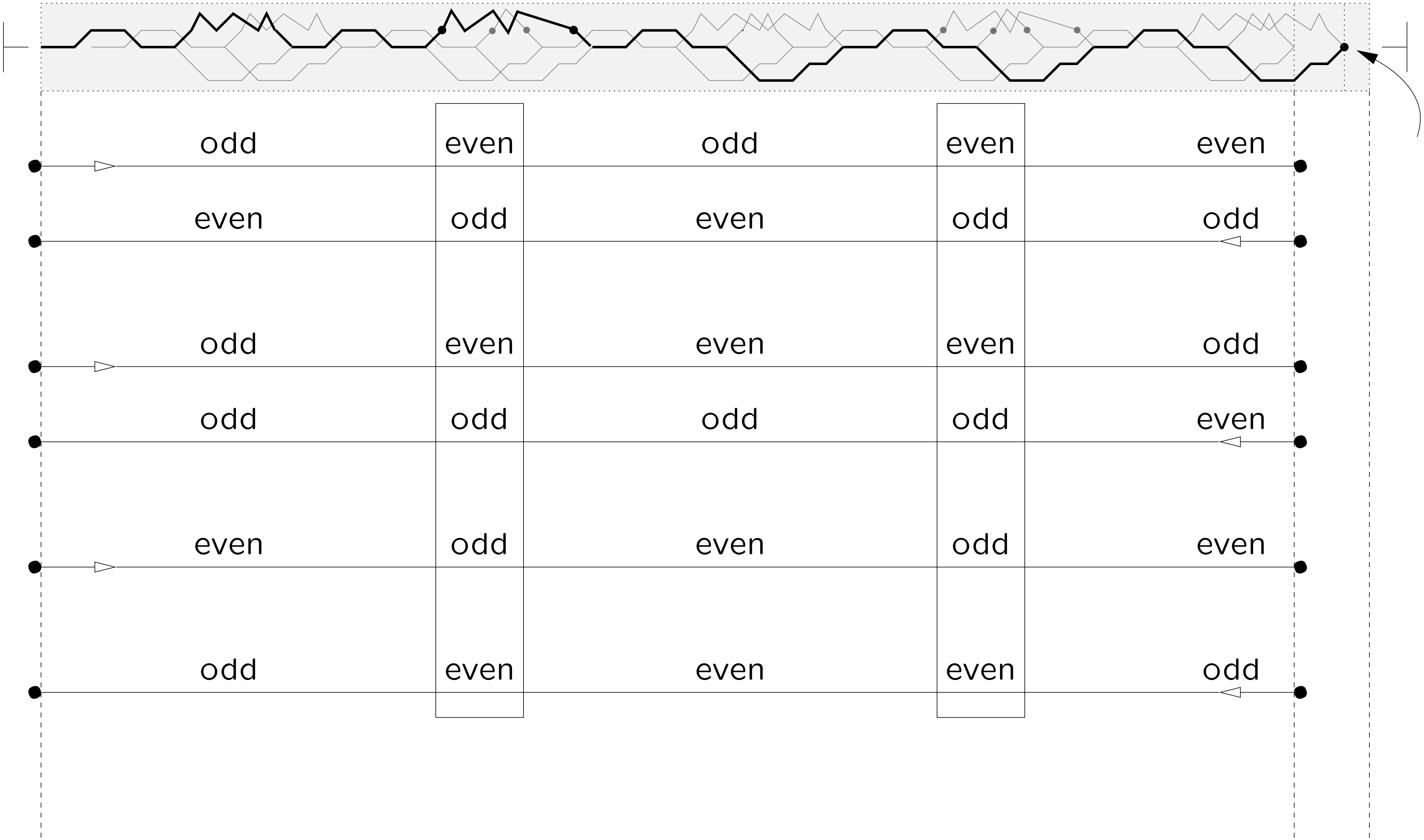

odd
even
odd
even
even
even
odd
even
odd
odd
odd
even
even
even
odd
odd
odd
odd
odd
even
even
odd
even
odd
even
odd
even
even
even
odd

the main argument

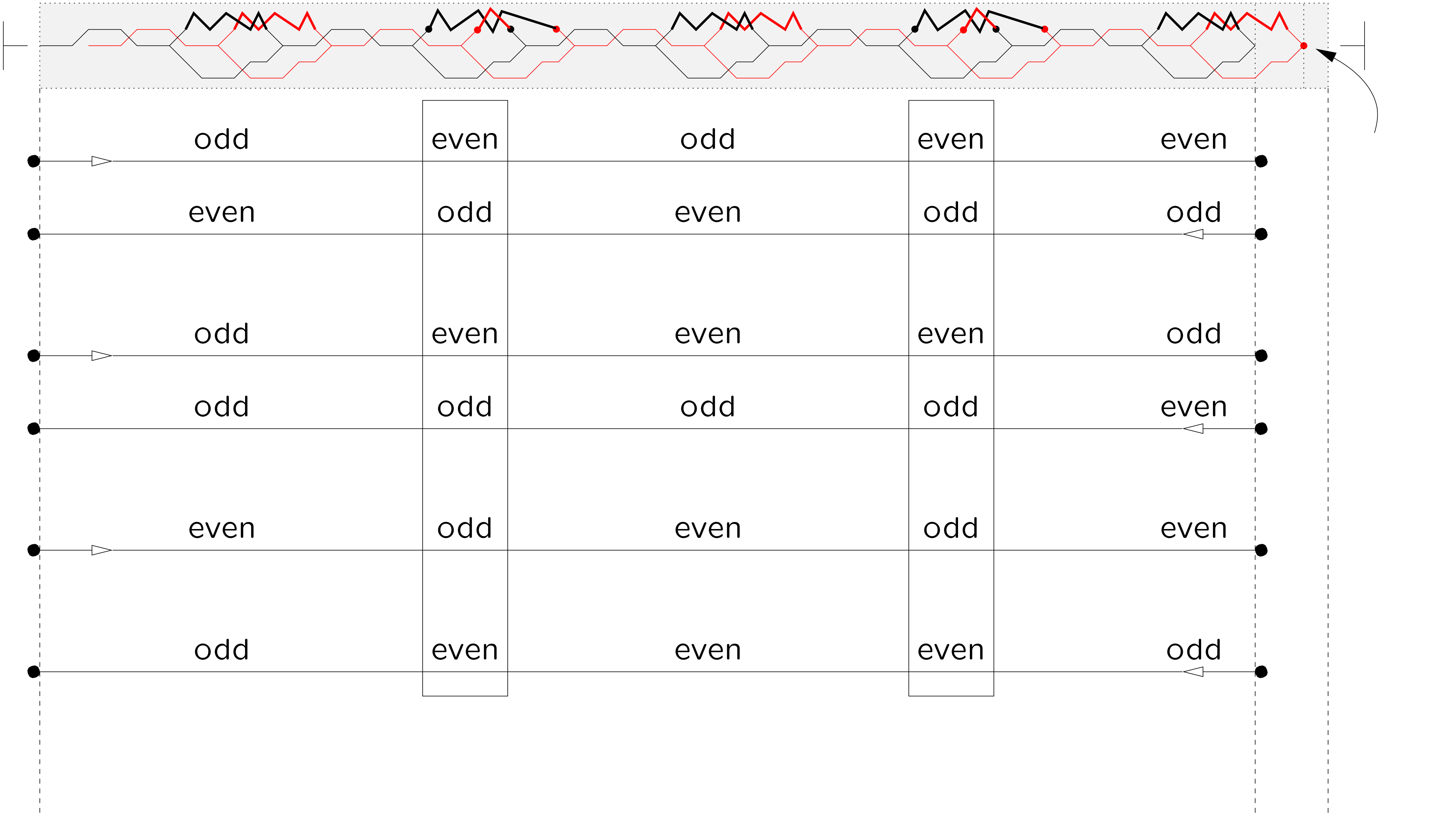
the main argument


odd
even
odd
even
even
even
odd
even
odd
odd
odd
even
even
even
odd
odd
odd
odd
odd
even
even
odd
even
odd
even
odd
even
even
even
odd

the main argument



the main argument



PROBLEM:	can small 2DFAs simulate small 1NFAs?
LIVENESS:	a complete problem for this conversion
MOLES:	a natural class of automata against liveness
GOAL:	show that **small** 2D moles cannot solve liveness
THEOREM:	even **huge** 2D moles cannot do it

PROBLEM:	can small 2DFAs simulate small 1NFAs?
LIVENESS:	a complete problem for this conversion
MOLES:	a natural class of automata against liveness
GOAL:	show that **small** 2D moles cannot solve liveness
THEOREM:	even **huge** 2D moles cannot do it

nice fact to know:

1. natural class of algorithms
2. the small 1NFAs for liveness are moles
3. unrestricted bidirectionality

PROBLEM:	can small 2DFAs simulate small 1NFAs?
LIVENESS:	a complete problem for this conversion
MOLES:	a natural class of automata against liveness
GOAL:	show that **small** 2D moles cannot solve liveness
THEOREM:	even **huge** 2D moles cannot do it

nice fact to know:

1. natural class of algorithms
2. the small 1NFAs for liveness are moles
3. unrestricted bidirectionality

BUT: *proof too technical*:

1. many technical details (skipped here)
2. hard to extend
3. hard to reuse its parts —but do check **dilemmas**

PROBLEM:	can small 2DFAs simulate small 1NFAs?
LIVENESS:	a complete problem for this conversion
MOLES:	a natural class of automata against liveness
GOAL:	show that **small** 2D moles cannot solve liveness
THEOREM:	even **huge** 2D moles cannot do it

nice fact to know:
1. natural class of algorithms
2. the small 1NFAs for liveness are moles
3. unrestricted bidirectionality

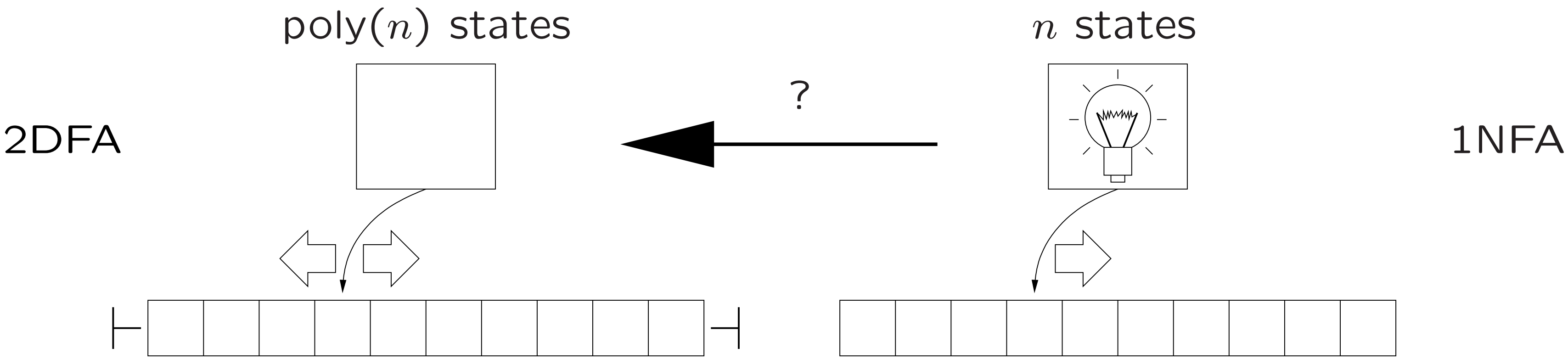
BUT: *proof too technical*:
1. many technical details (skipped here)
2. hard to extend
3. hard to reuse its parts —but do check **dilemmas**

class too restricted:
1. computability answer to a complexity question
2. we have definitely missed the real reasons. . .

why care about this problem

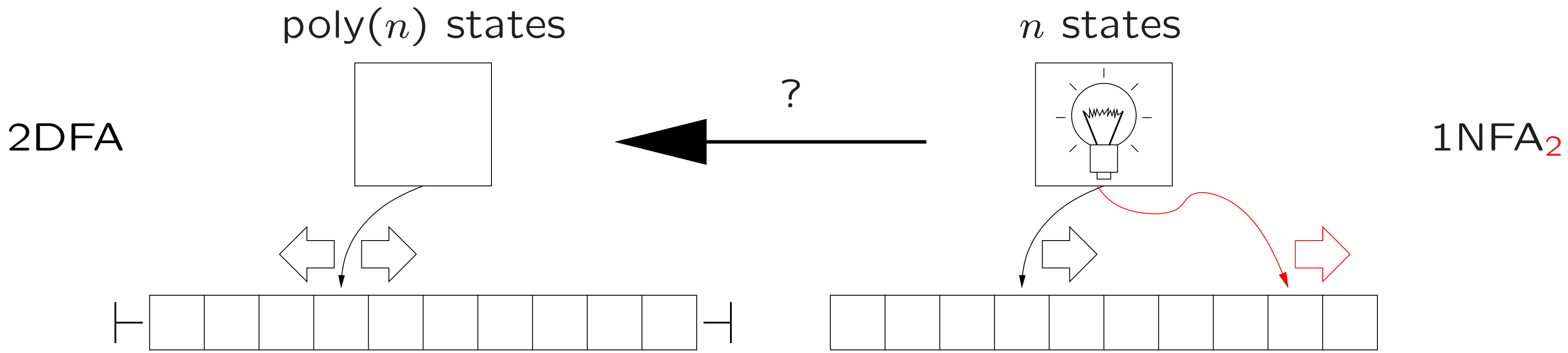
REASON #1:

REASON #2:



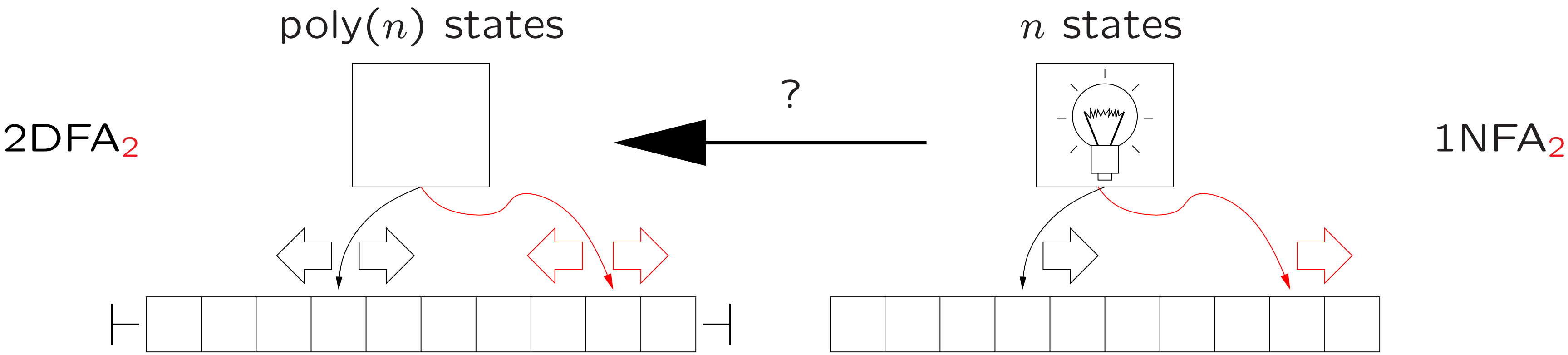
REASON #1:

REASON #2:



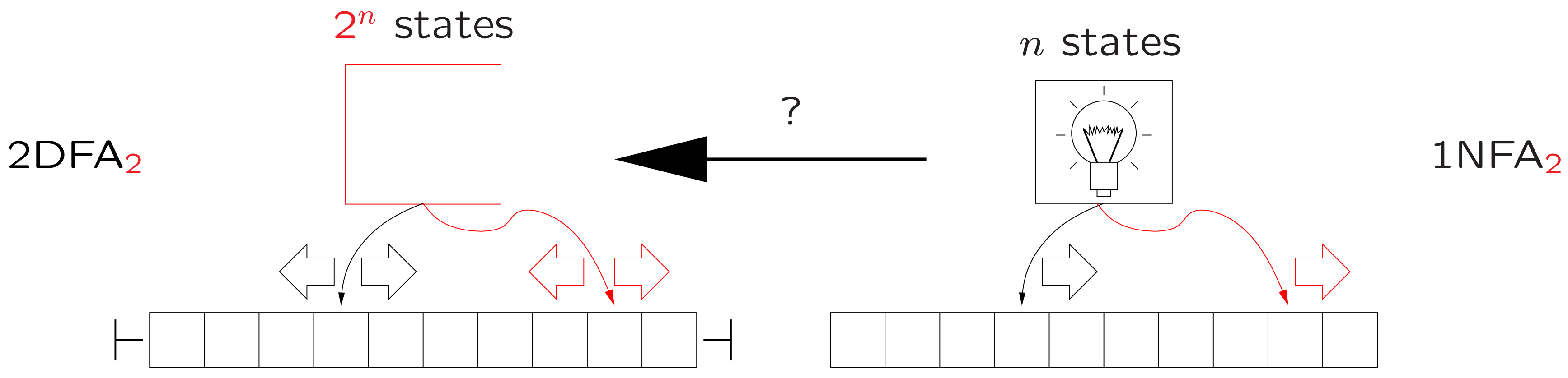
REASON #1:

REASON #2:



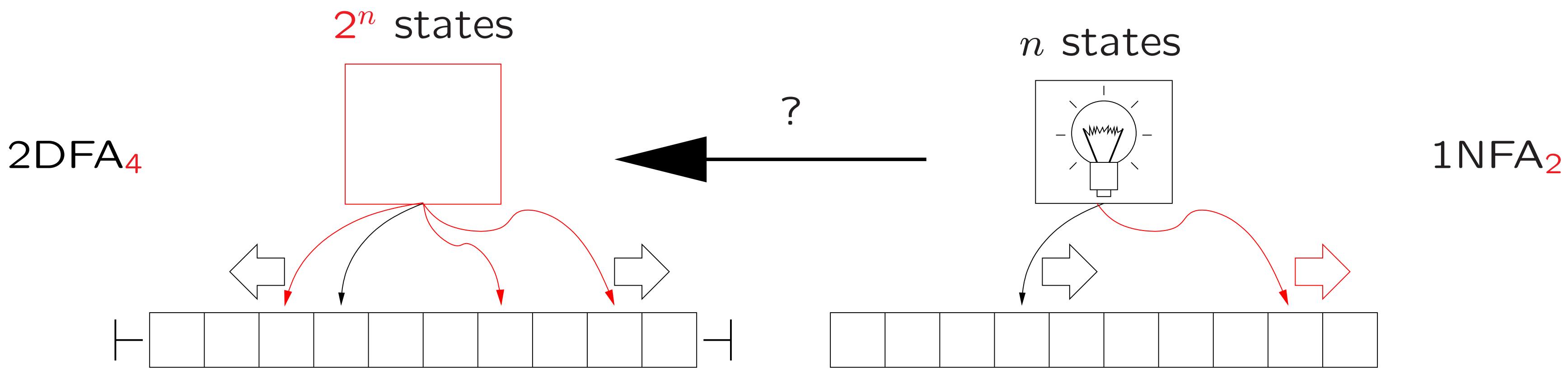
REASON #1:

REASON #2:



REASON #1:

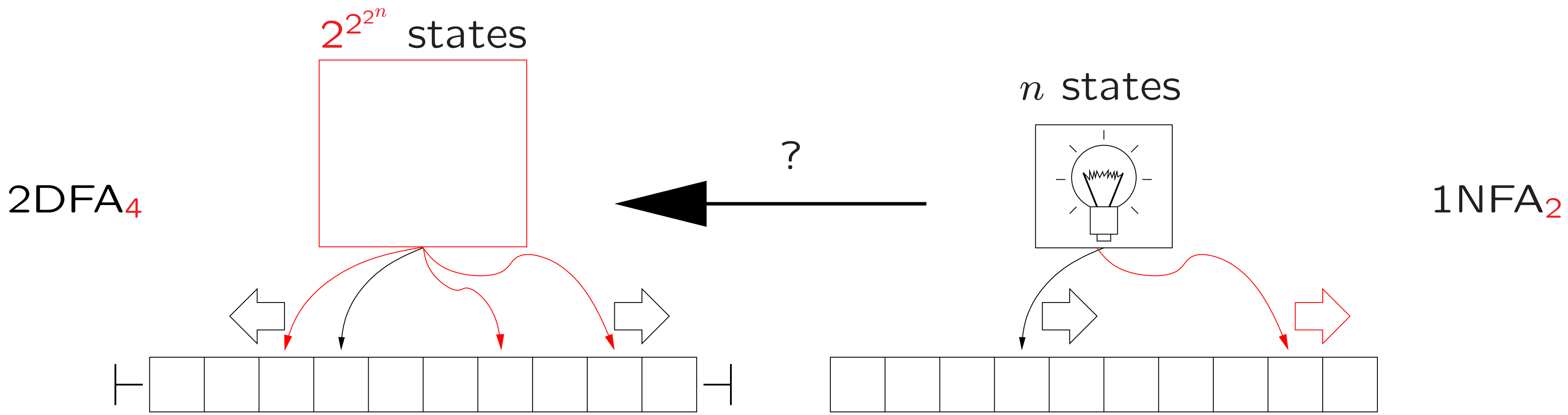
REASON #2:



why care about this problem

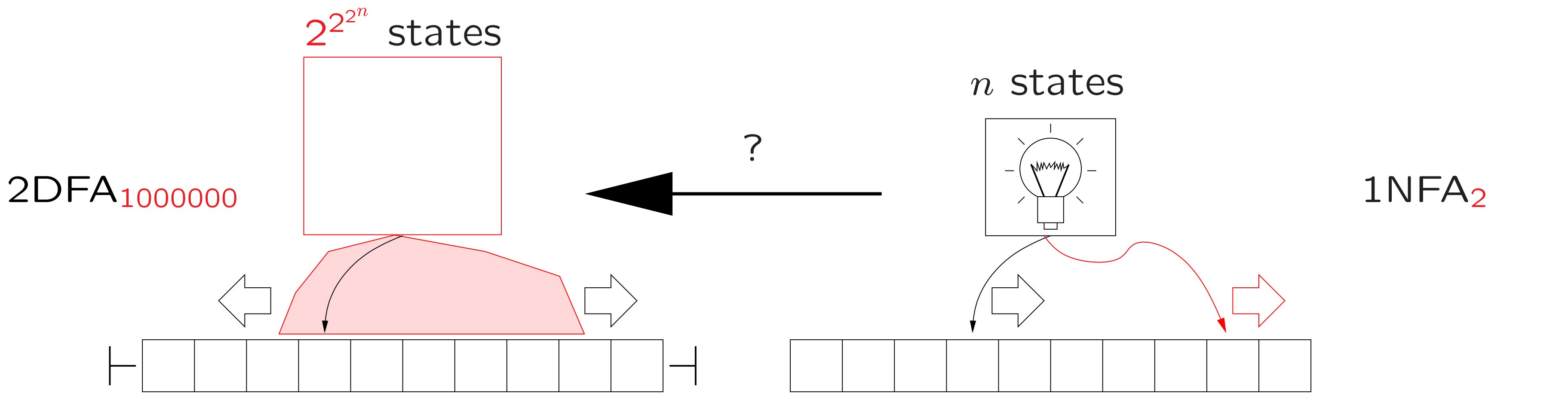
REASON #1:

REASON #2:



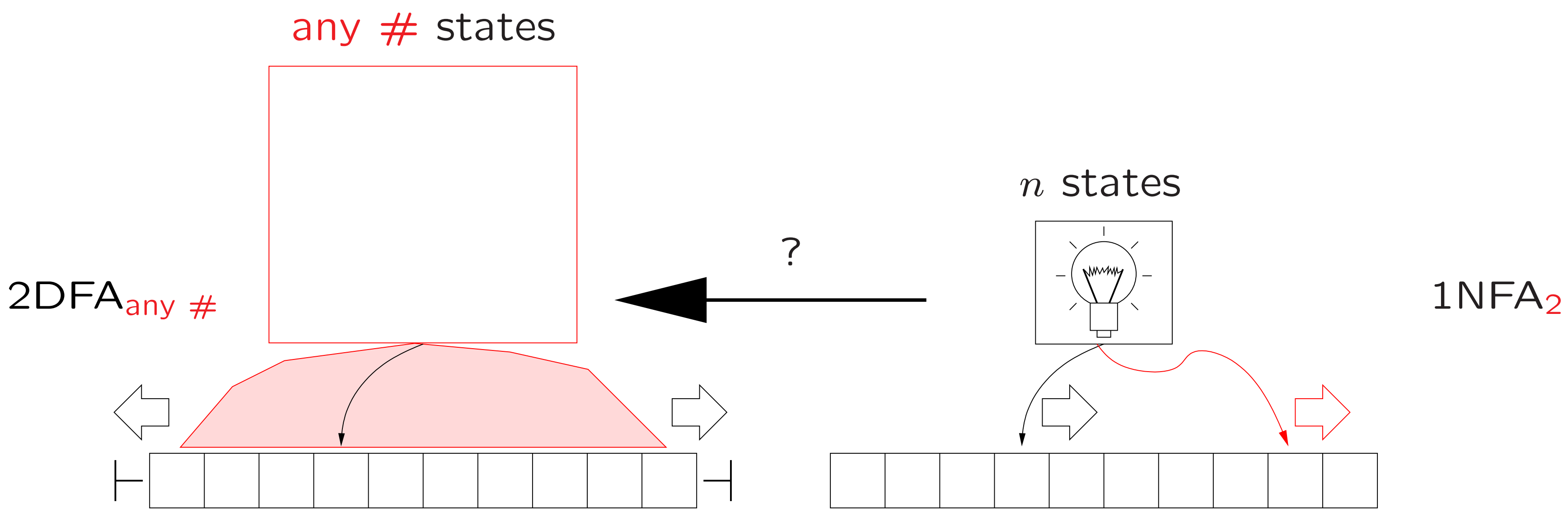
REASON #1:

REASON #2:



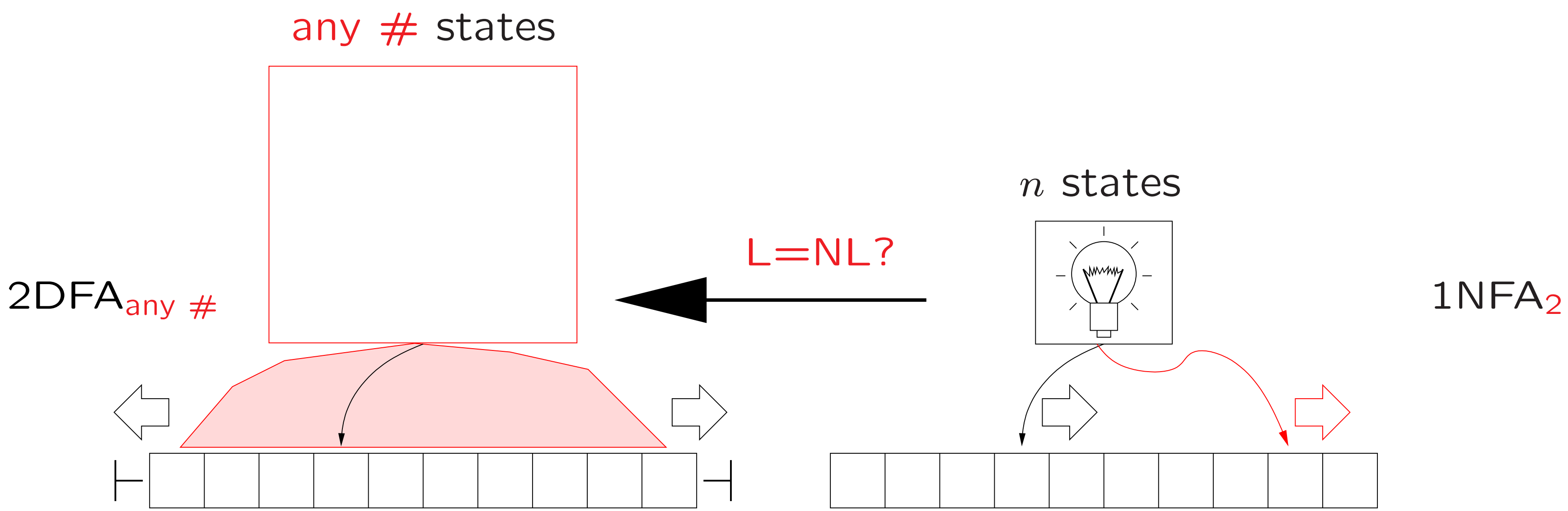
REASON #1:

REASON #2:



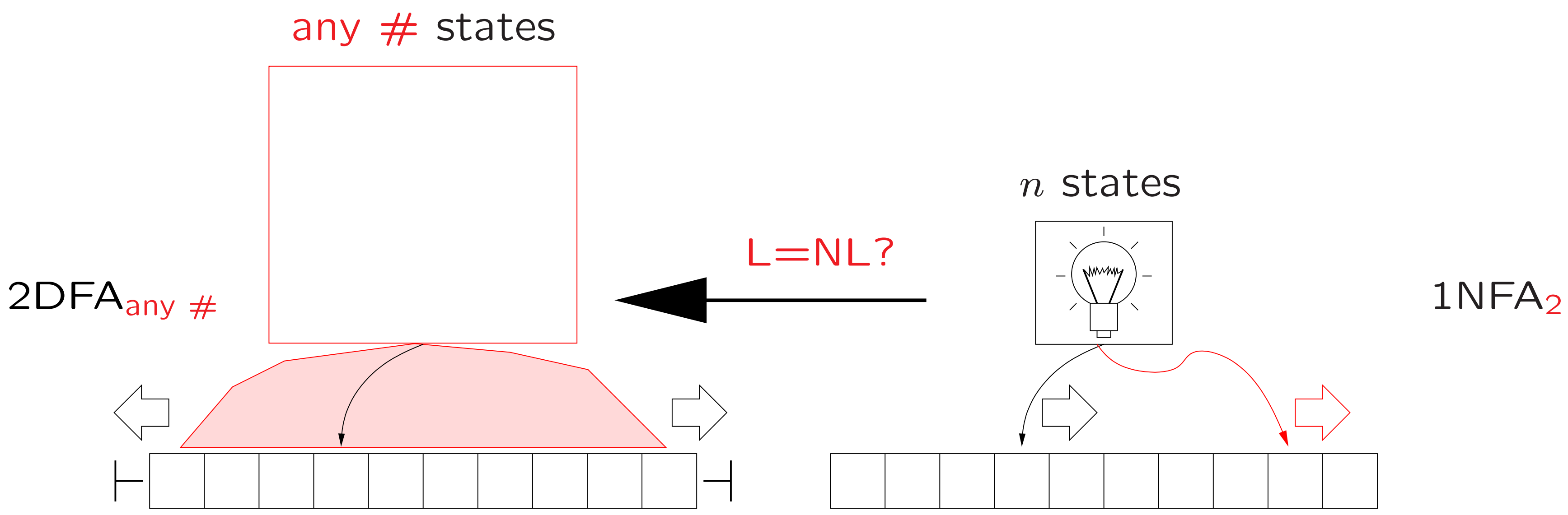
REASON #1:

REASON #2:



REASON #1:

REASON #2: two-way determinism vs. one-way nondeterminism



REASON #1: it is such a nice problem!

REASON #2: two-way determinism vs. one-way nondeterminism