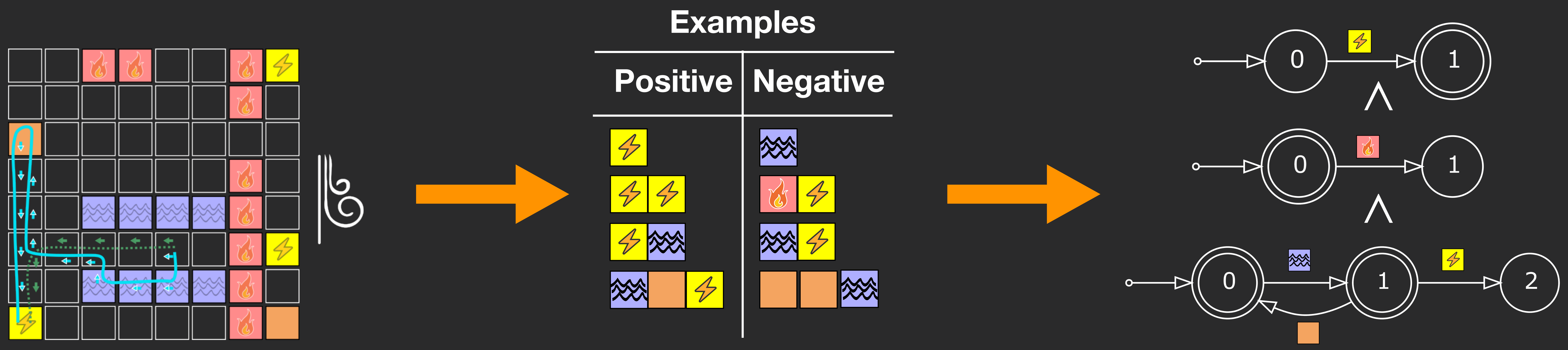
Learning DFA Decompositions from Examples and Demonstrations

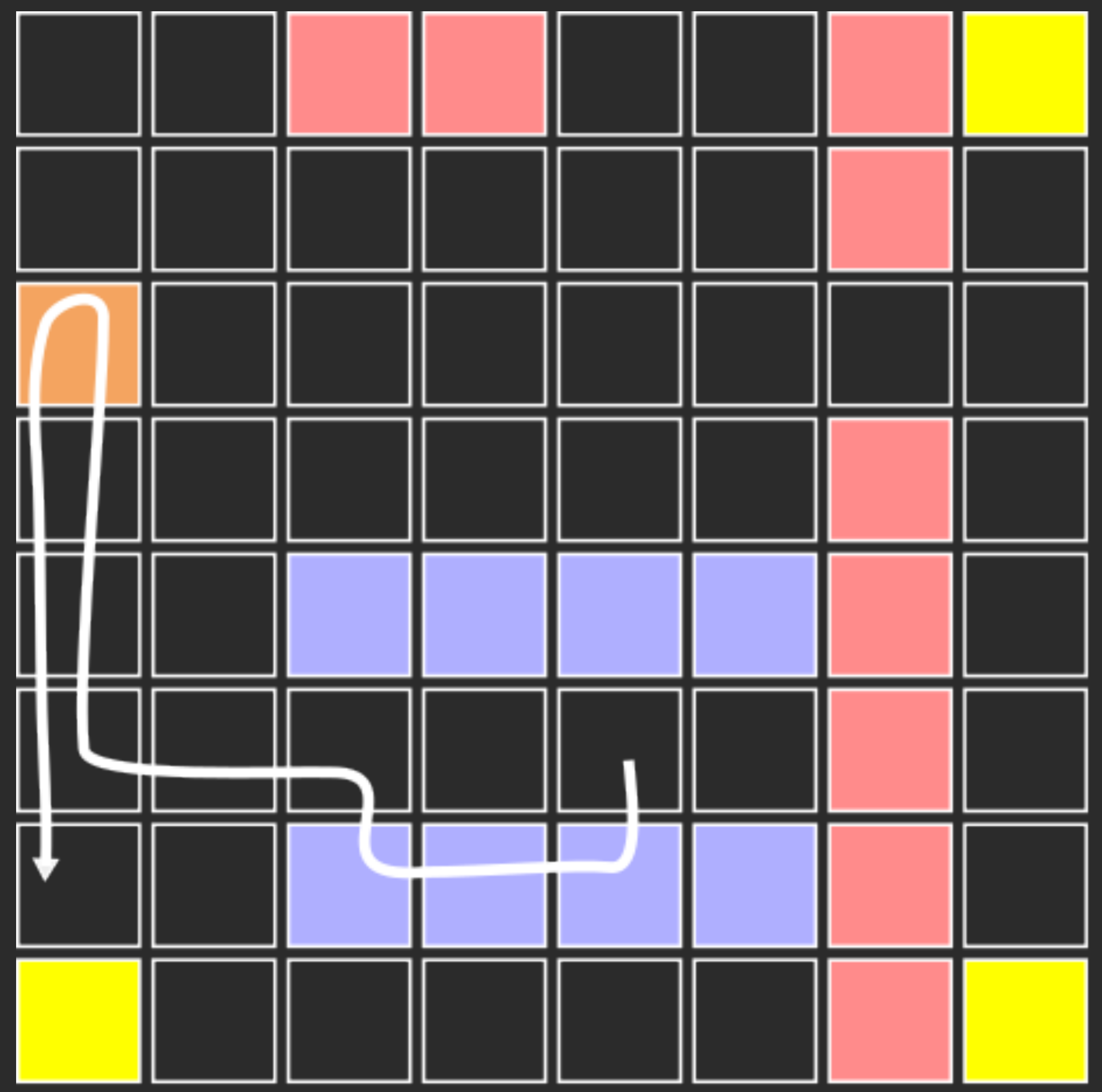


Niklas Lauffer*, Beyazit Yalcinkaya*, Marcell Vazquez-Chanlatte, Ameesh Shah, and Sanjit A. Seshia

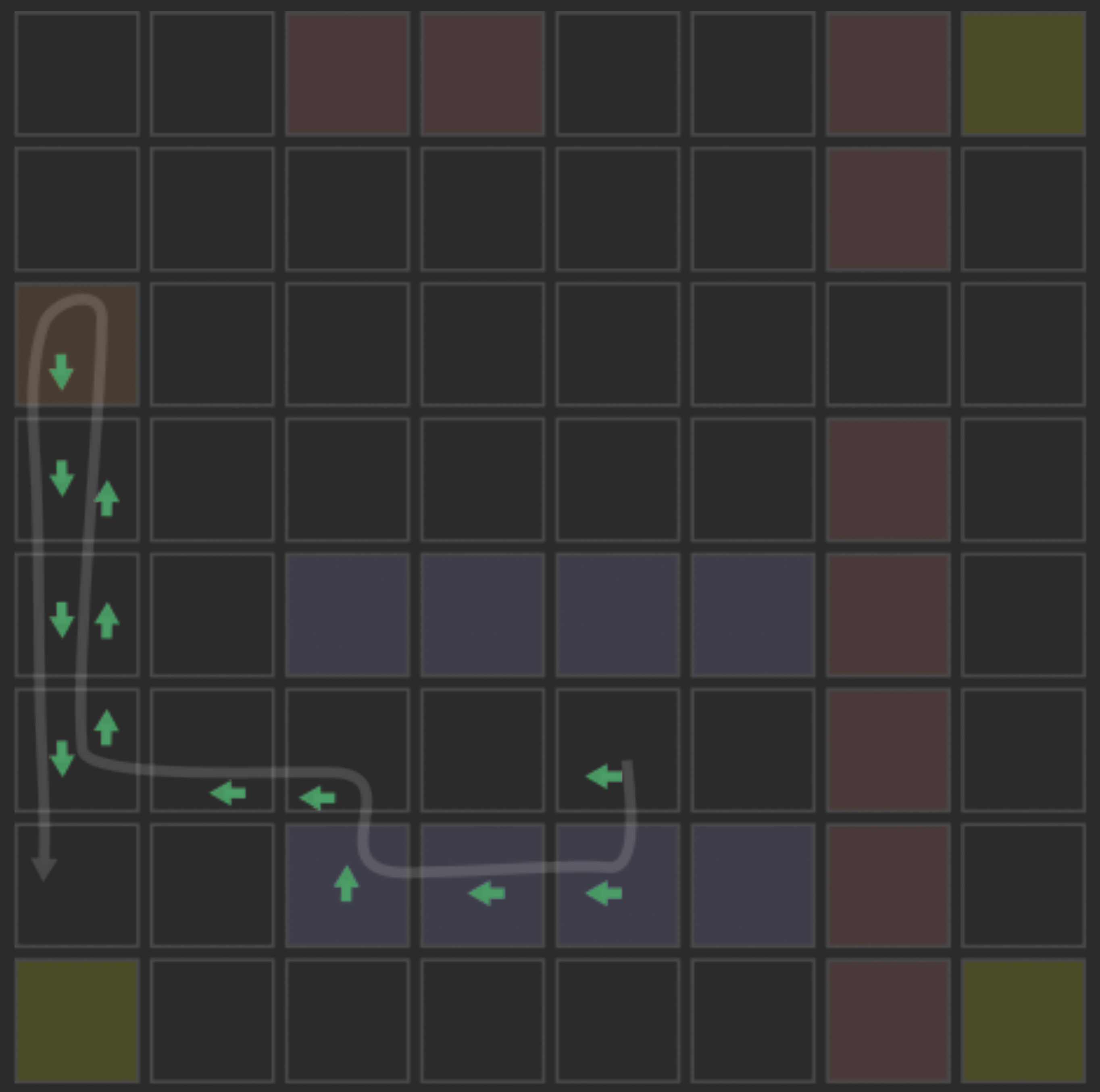
*equal contribution

University of California, Berkeley

A simple gridworld

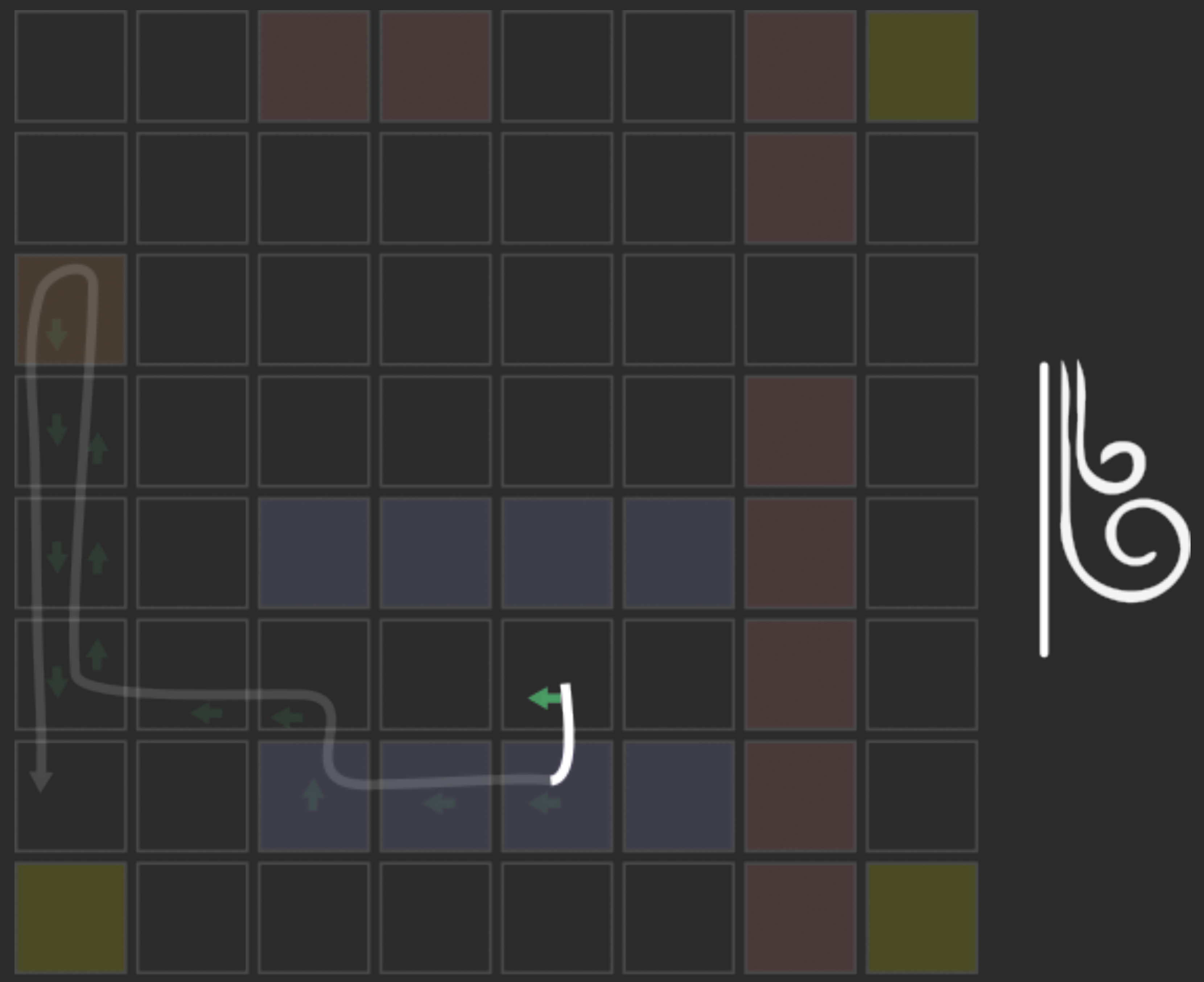


A simple gridworld



Actions = {↑ ↓ ← →}

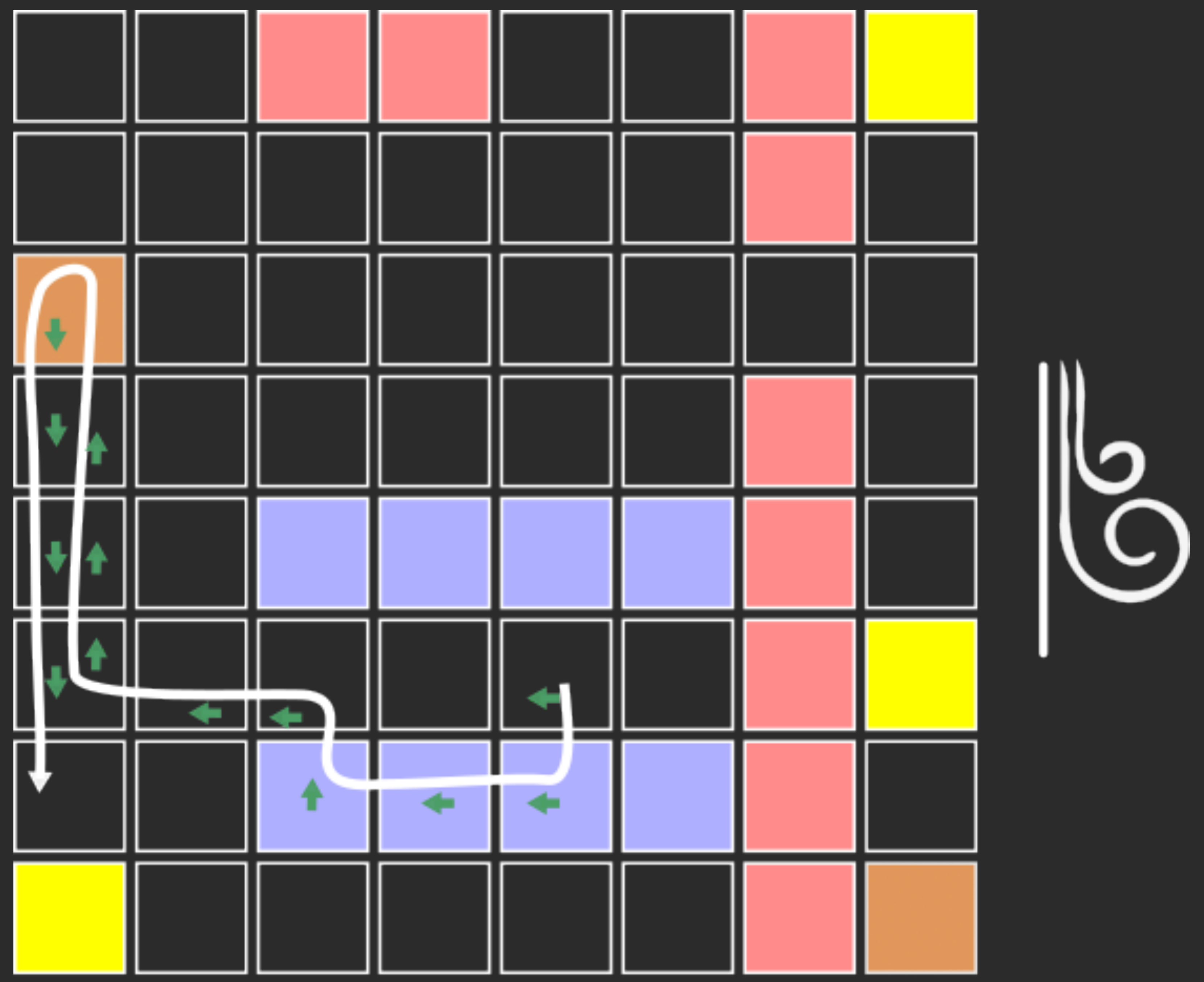
A simple gridworld



Actions = $\{\uparrow \downarrow \leftarrow \rightarrow\}$

Pr(slip $\downarrow$) = 1/32

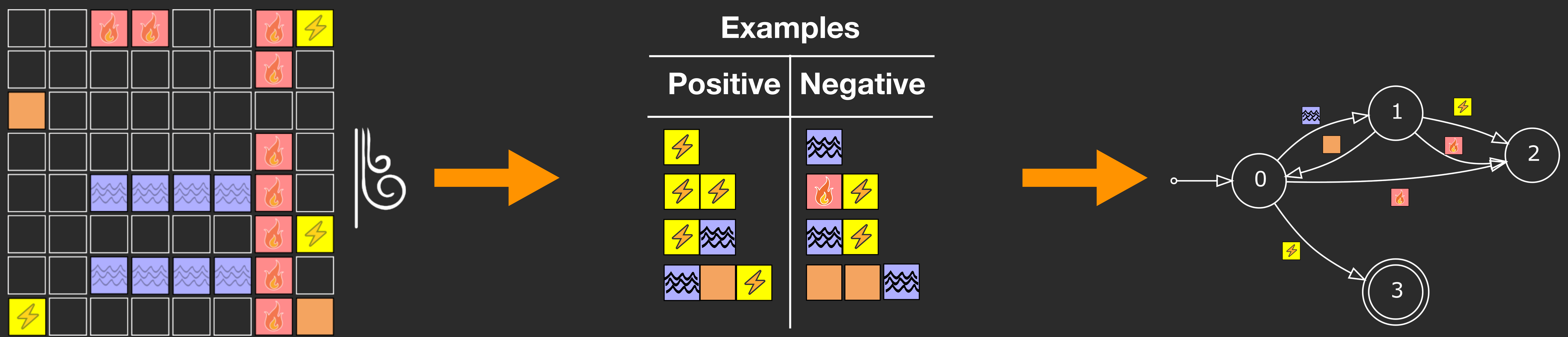
A simple gridworld



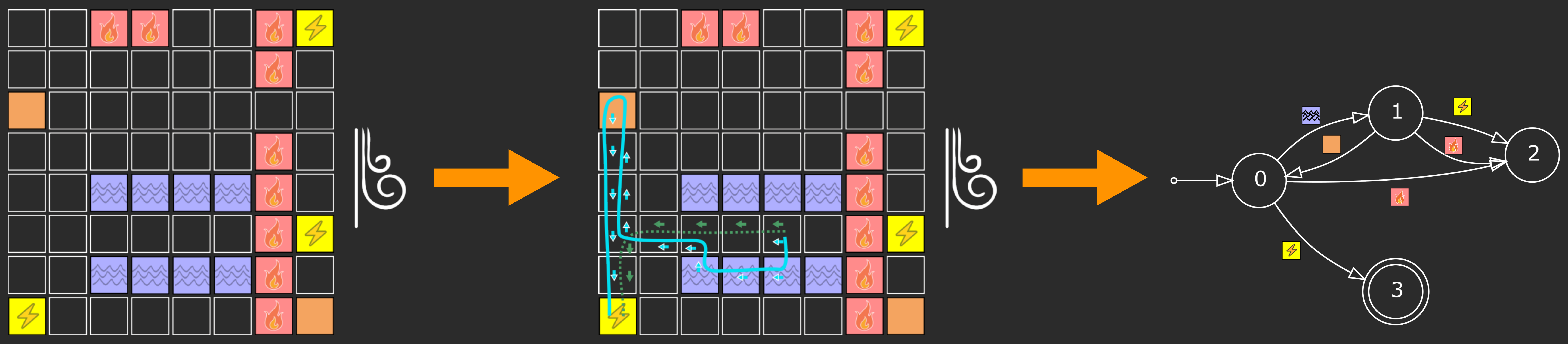
Actions = {↑ ↓ ← →}

Pr(slip ↓) = 1/32

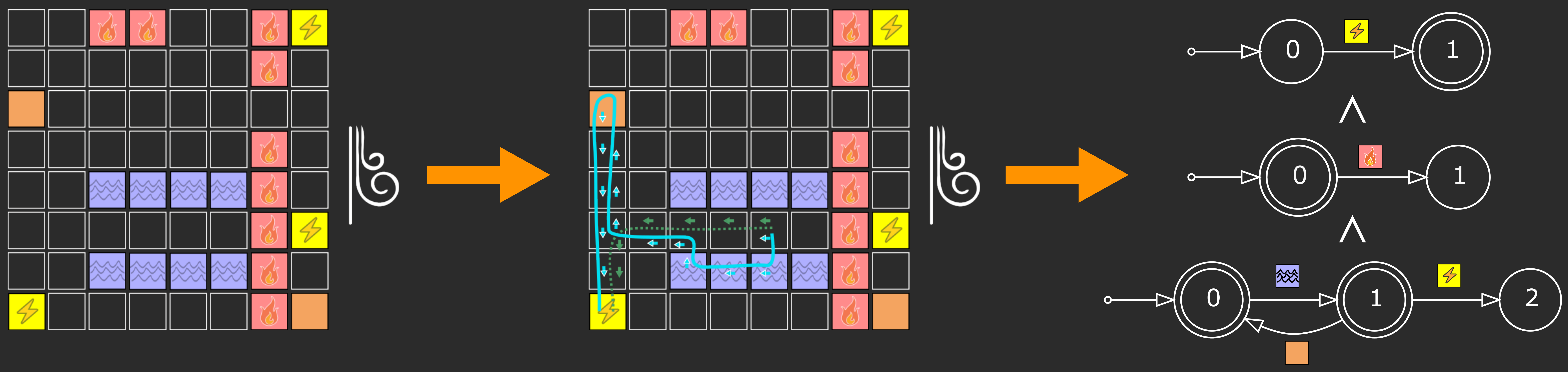
Specification mining



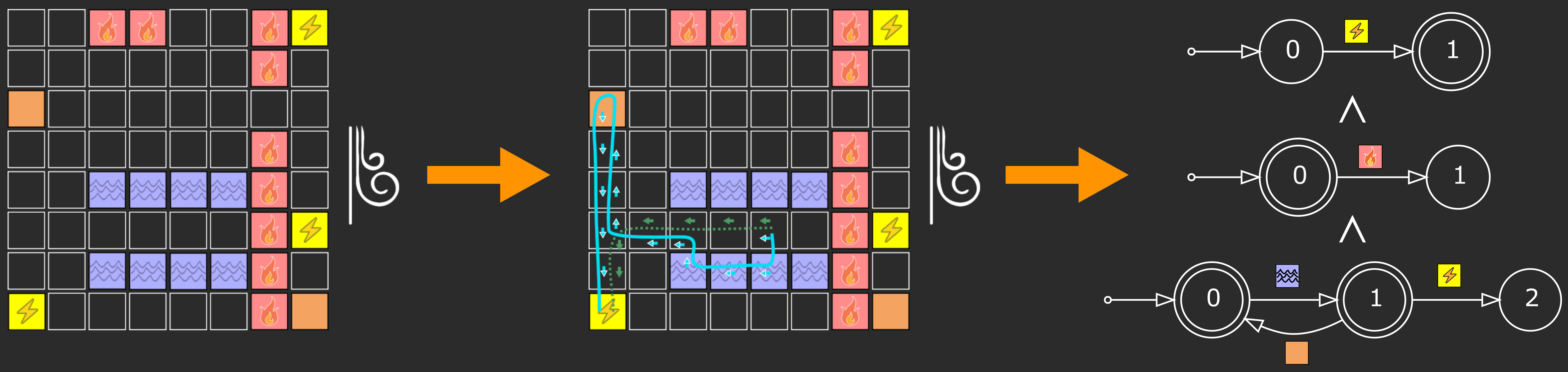
Learning from demonstrations



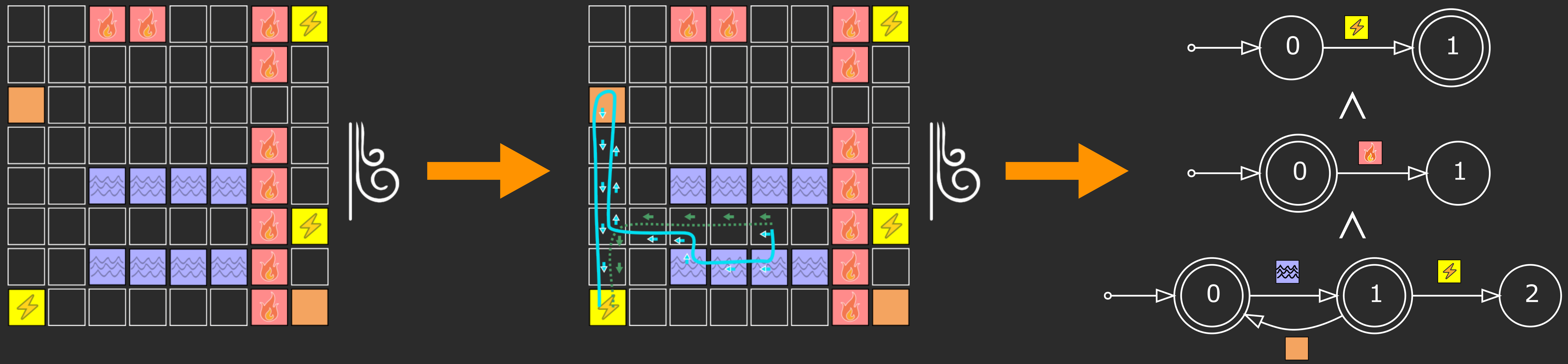
Learning decompositions



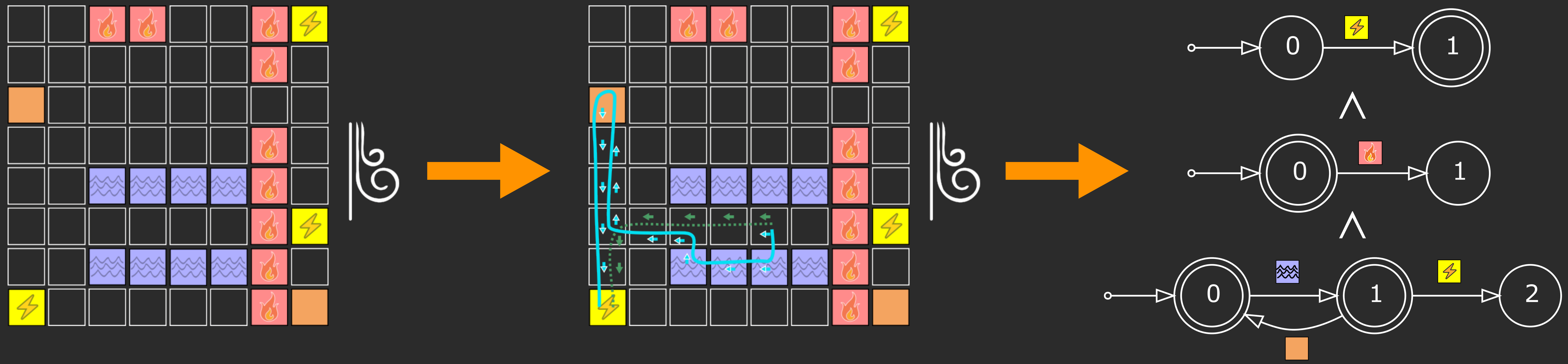
Monolithic specifications can often be difficult to understand



System-level specifications are often conjunctions of sub-specifications



Inductive bias matters when learning from few demonstrations



Contributions

1. SAT-based encoding for identifying a DFA decomposition of a specific size from labeled examples

Contributions

1. SAT-based encoding for identifying a DFA decomposition of a specific size from labeled examples

2. An algorithm for enumerating the full Pareto-frontier of decompositions

Contributions

1. SAT-based encoding for identifying a DFA decomposition of a specific size from labeled examples

2. An algorithm for enumerating the full Pareto-frontier of decompositions

3. Experimental analysis and extension to learning from demonstrations

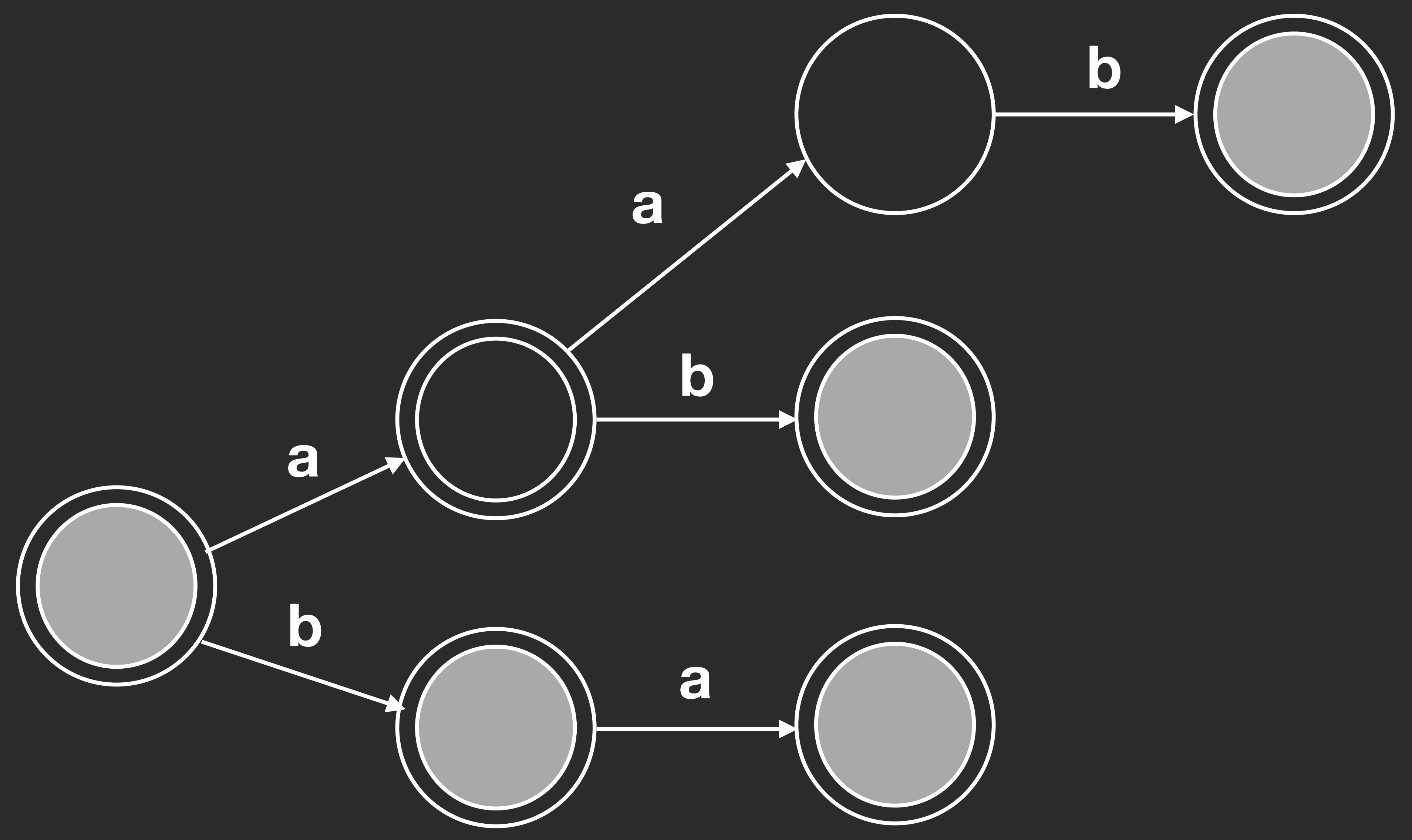
Structure of the talk

1. Technical details

2. Scalability analysis

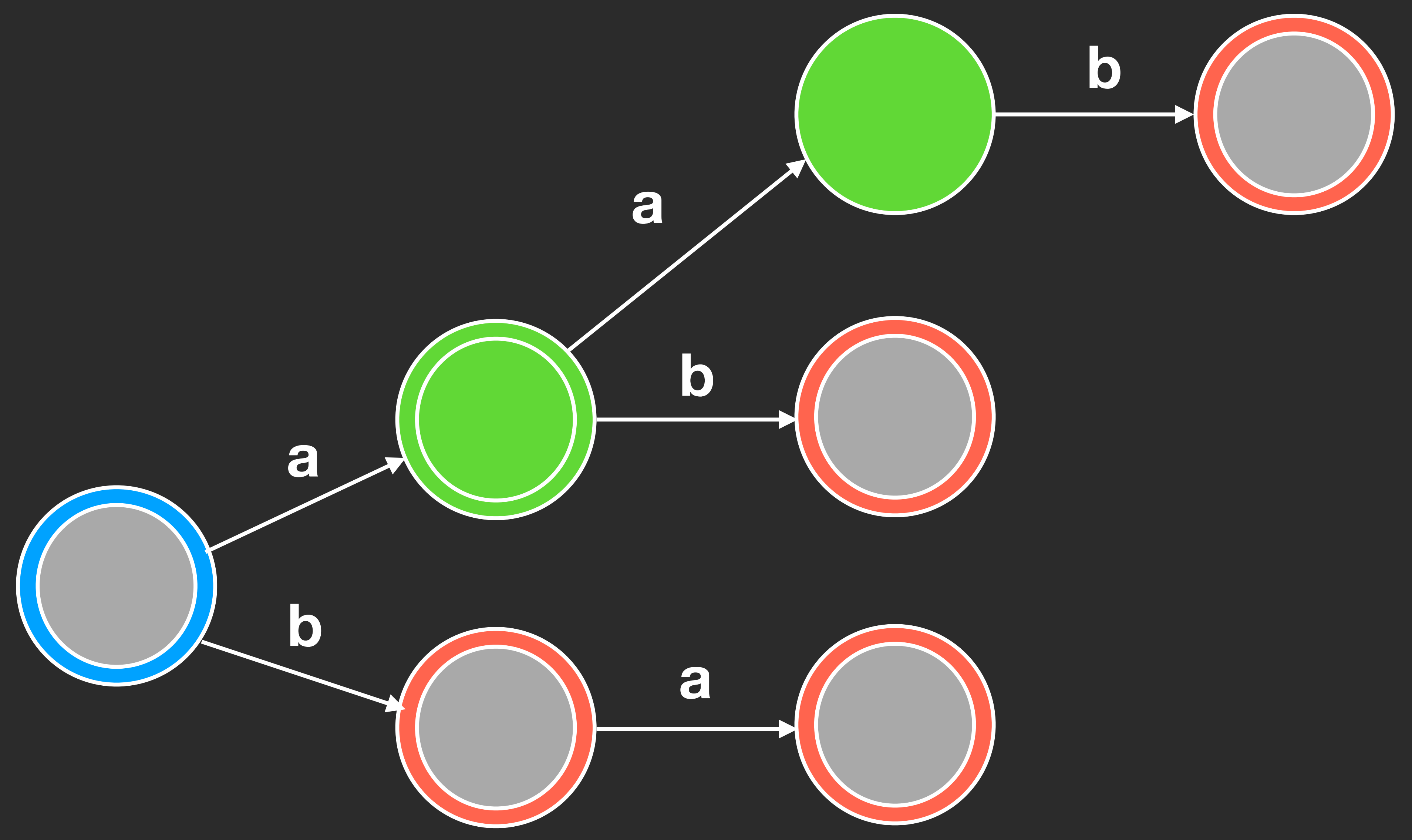
3. Learning from demonstrations

State merging via coloring



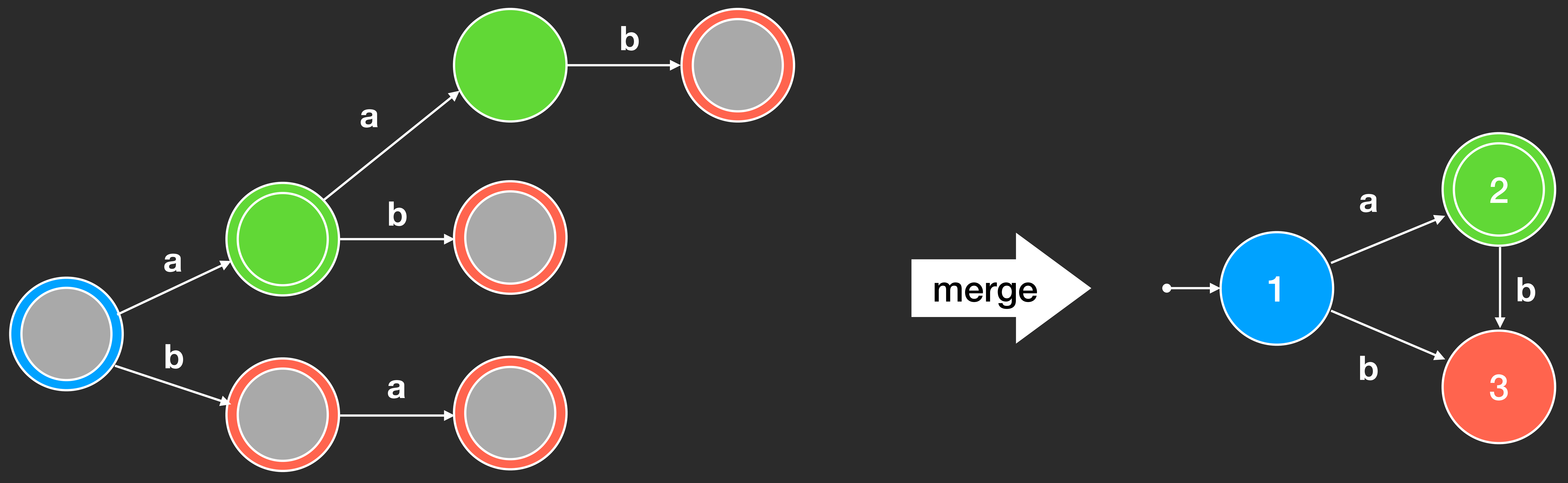
Positive: {a} **Negative: {Ø, b, ba, ab, aab}**

State merging via coloring



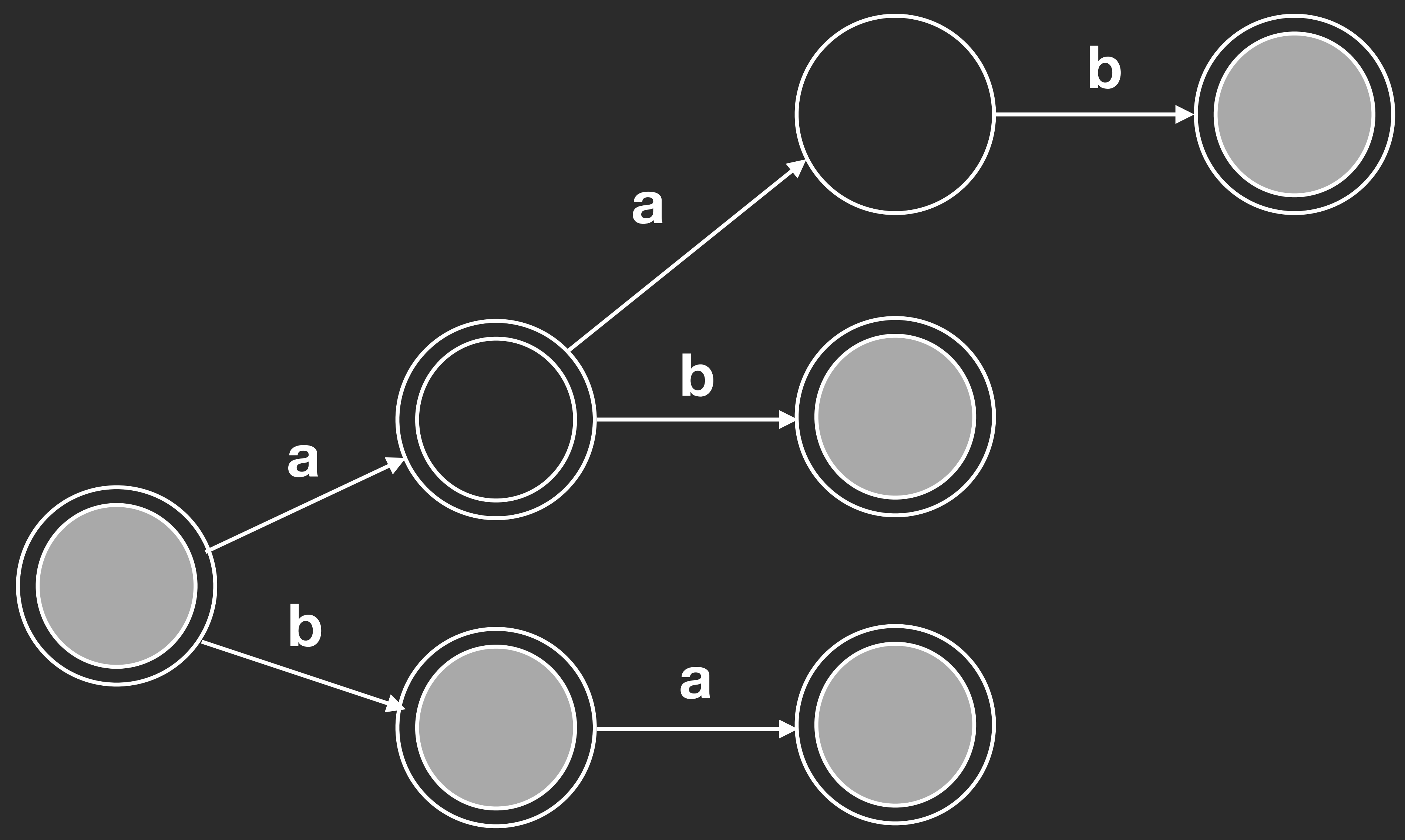
Positive: {a} Negative: {Ø, b, ba, ab, aab}

State merging via coloring



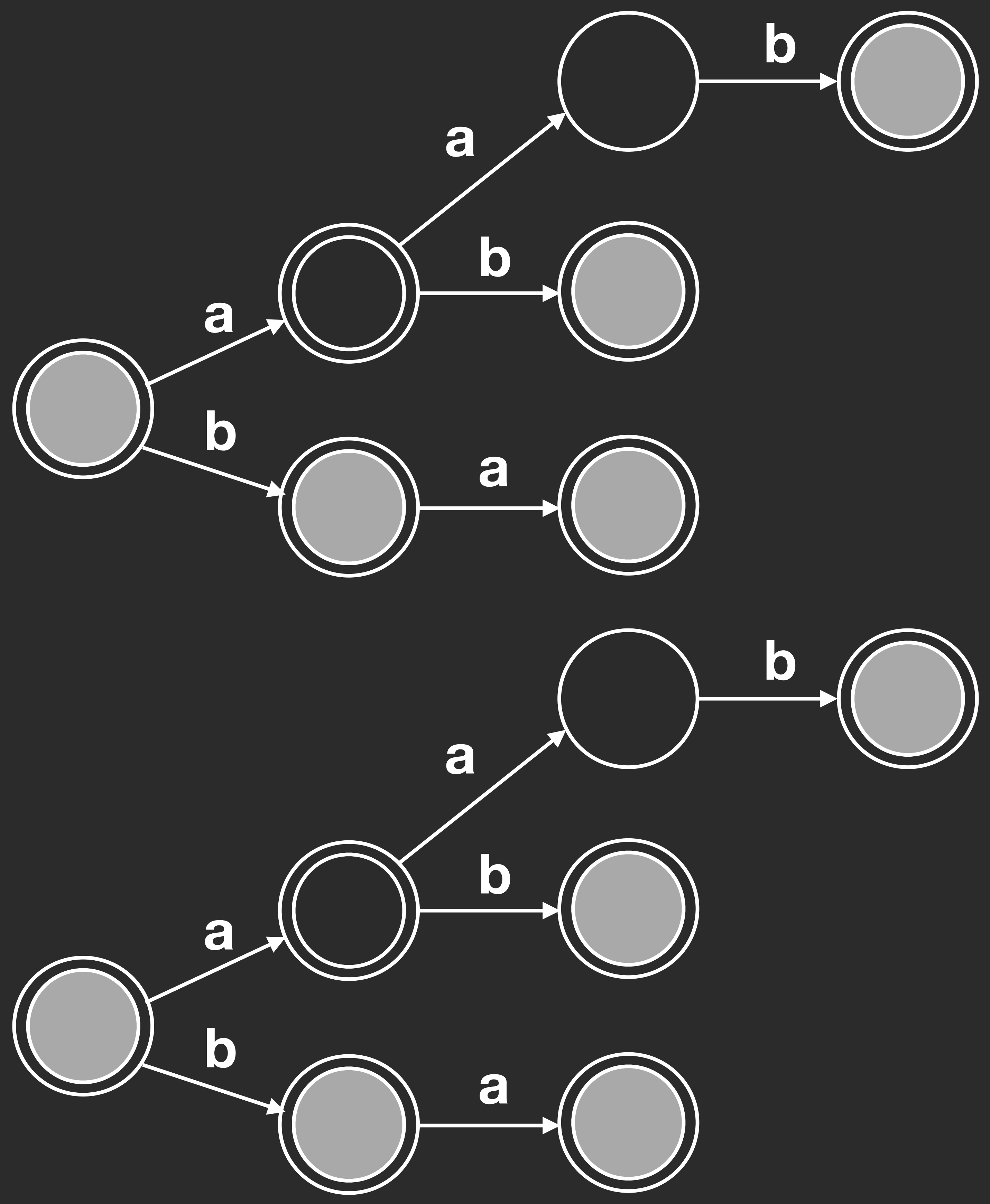
Positive: {a} Negative: {Ø, b, ba, ab, aab}

State merging for decompositions



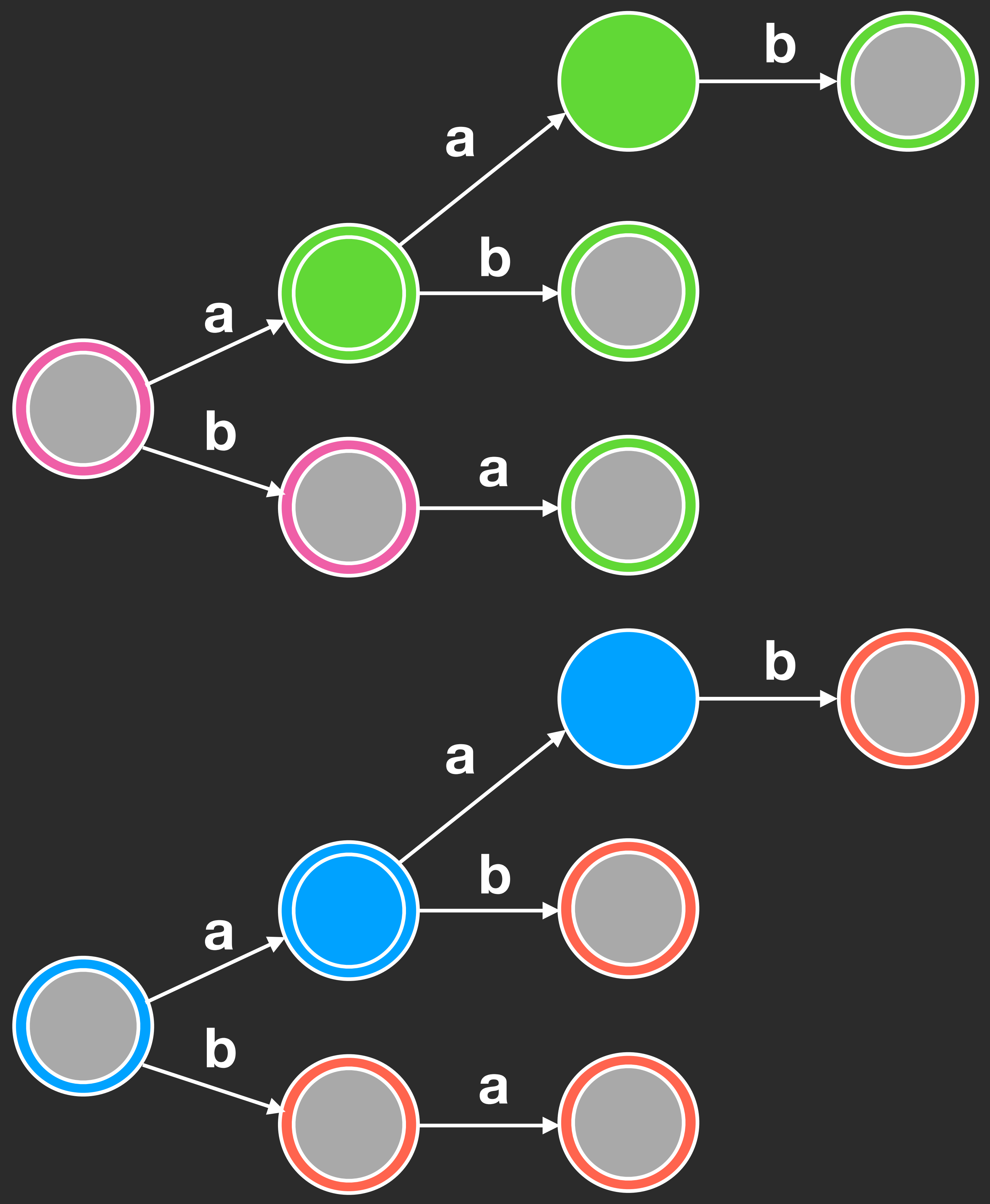
Positive: {a} **Negative: {Ø, b, ba, ab, aab}**

State merging for decompositions



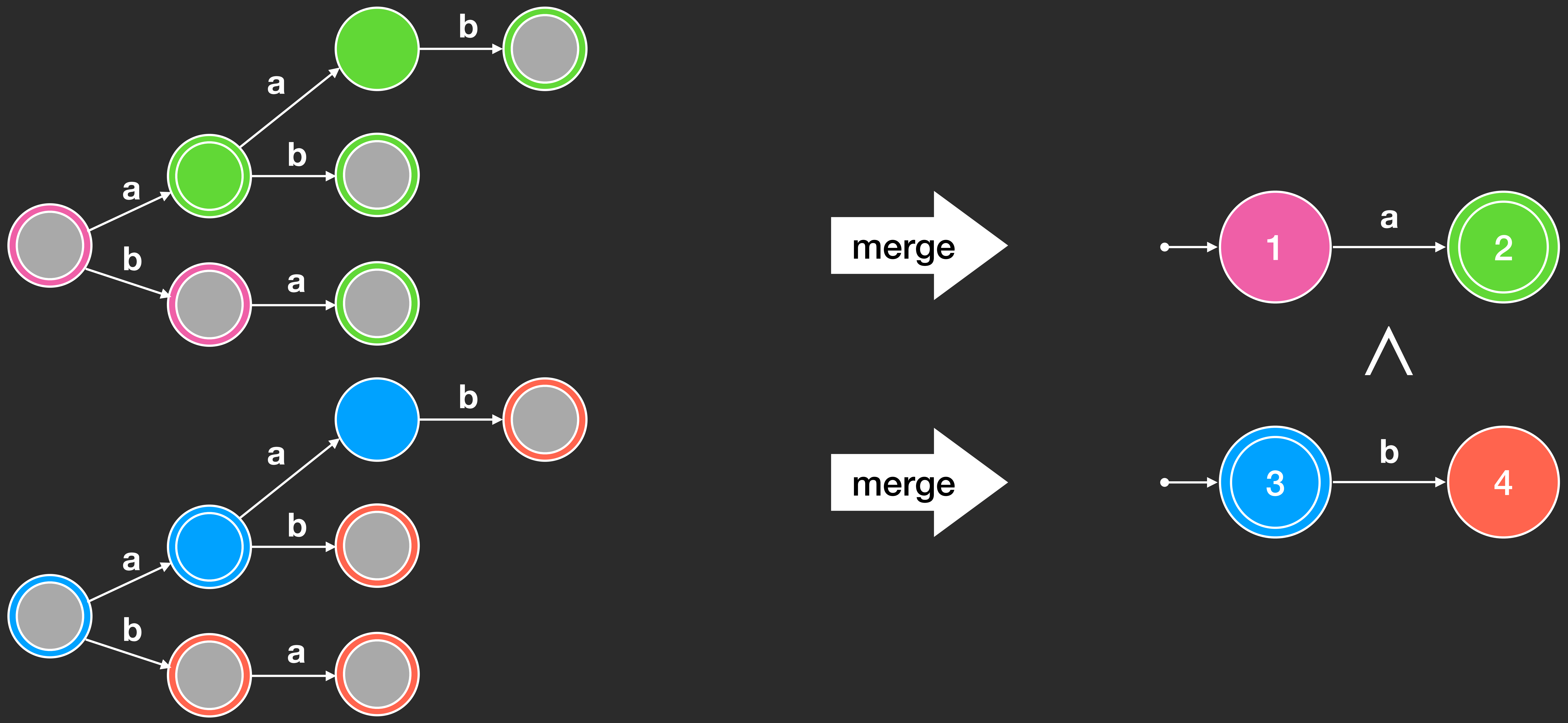
Positive: {a} **Negative: {Ø, b, ba, ab, aab}**

State merging for decompositions



Positive: {a} **Negative: {Ø, b, ba, ab, aab}**

State merging for decompositions



Positive: {a} **Negative: {Ø, b, ba, ab, aab}**

A SAT encoding

Implemented as an extension of existing work*

*Ulyantsev, Vladimir & Zakirzyanov, Ilya & Shalyto, Anatoly. (2015). BFS-based Symmetry Breaking Predicates for DFA Identification

A SAT encoding

Implemented as an extension of existing work*

- Each negative example must be rejected by *at least* one DFA:

$$\bigwedge_{v \in V_-} \bigvee_{k \in [n]} \bigwedge_{i \in [m_k]} x^k_{v,i} \implies \neg z^k_i \,.$$

*Ulyantsev, Vladimir & Zakirzyanov, Ilya & Shalyto, Anatoly. (2015). BFS-based Symmetry Breaking Predicates for DFA Identification

A SAT encoding

Implemented as an extension of existing work*

- Each negative example must be rejected by *at least* one DFA:

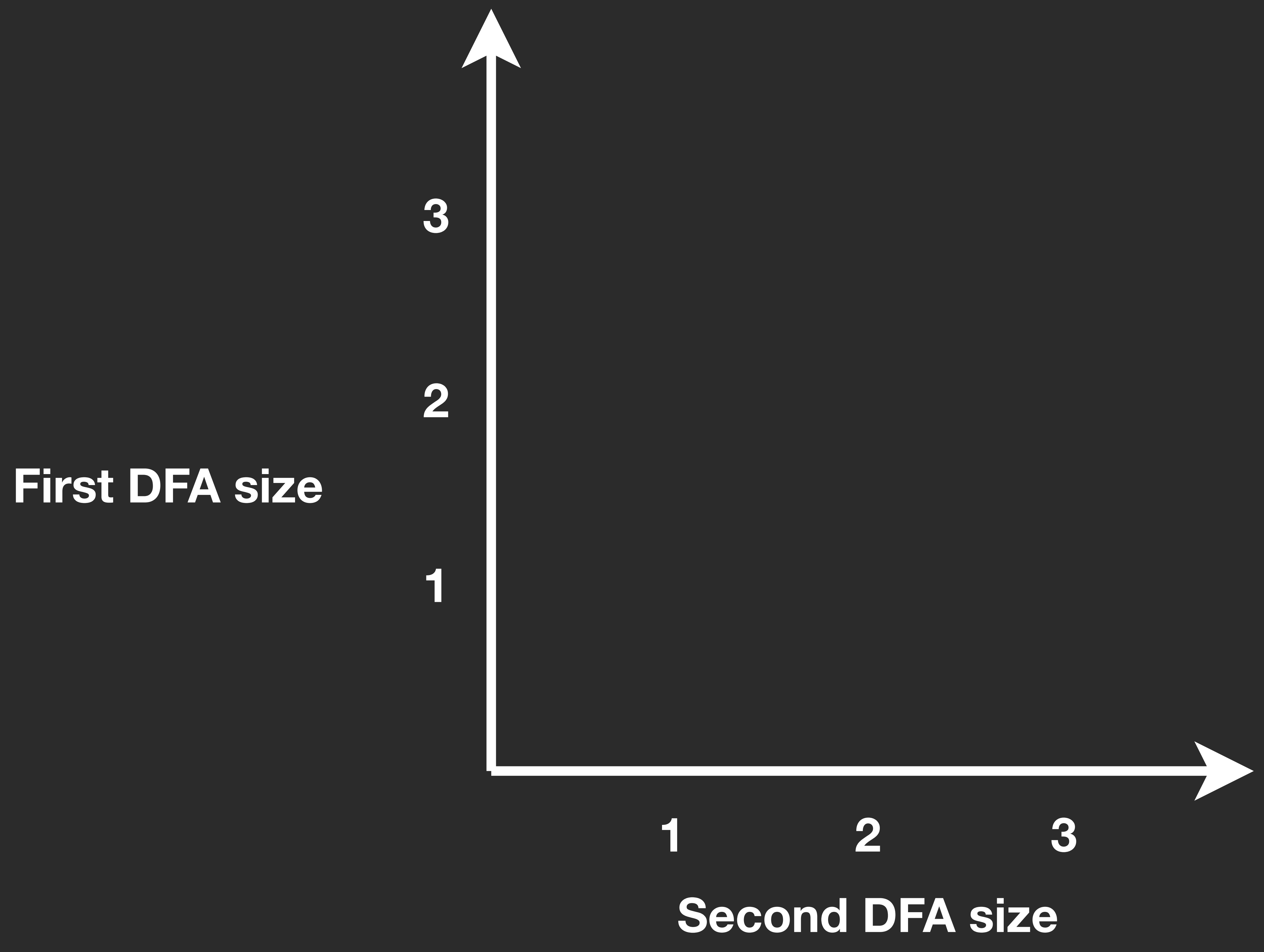
$$\bigwedge_{v \in V_-} \bigvee_{k \in [n]} \bigwedge_{i \in [m_k]} x^k_{v,i} \implies \neg z^k_i .$$

- Accepting and rejecting states of *individual* prefix trees cannot be merged:

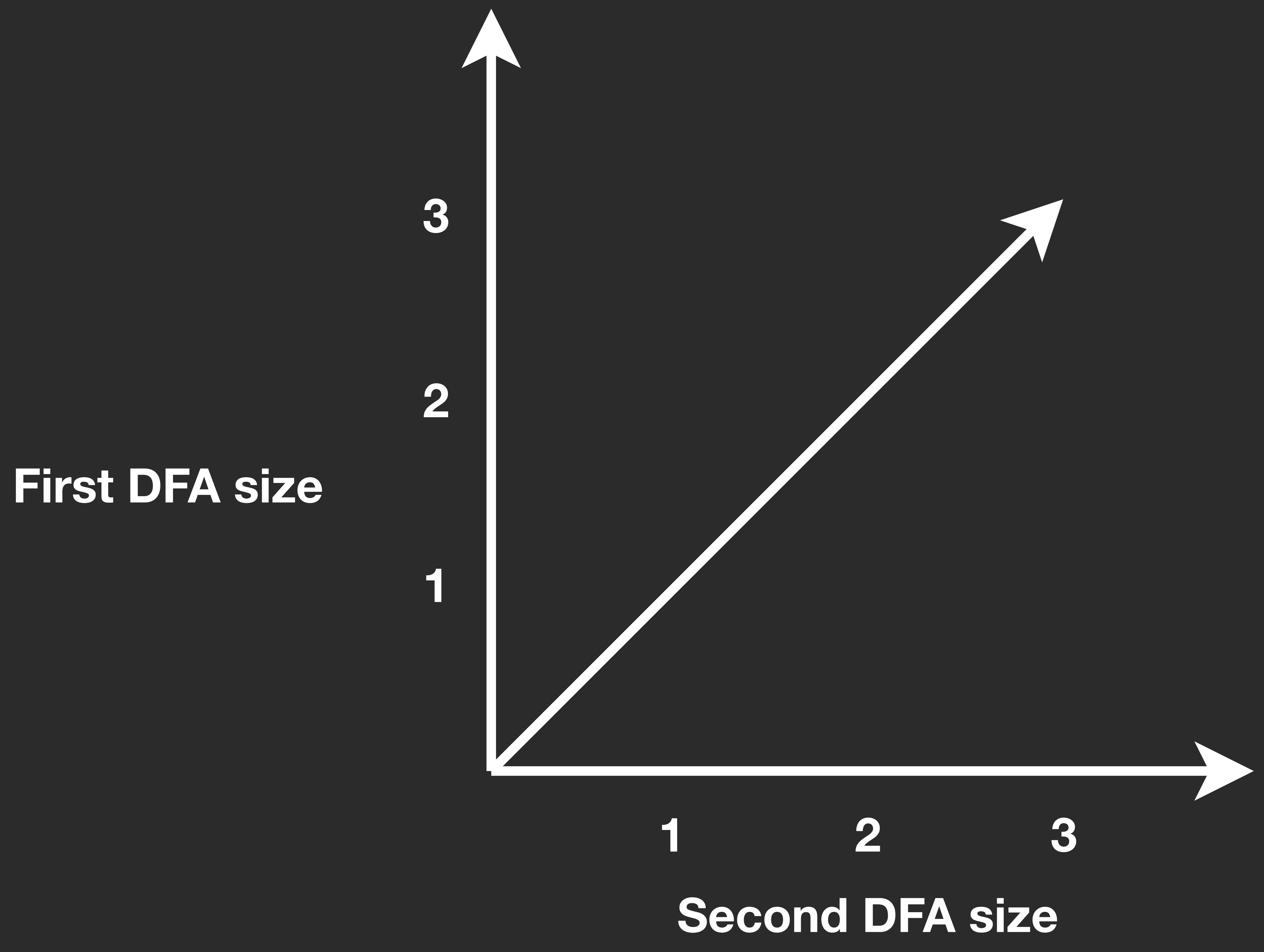
$$\bigwedge_{v_- \in V_-} \bigwedge_{v_+ \in V_+} \bigwedge_{k \in [n]} \bigwedge_{i \in [m_k]} (x^k_{v_-,i} \land \neg z^k_i) \implies \neg x^k_{v_+,i}$$

*Ulyantsev, Vladimir & Zakirzyanov, Ilya & Shalyto, Anatoly. (2015). BFS-based Symmetry Breaking Predicates for DFA Identification

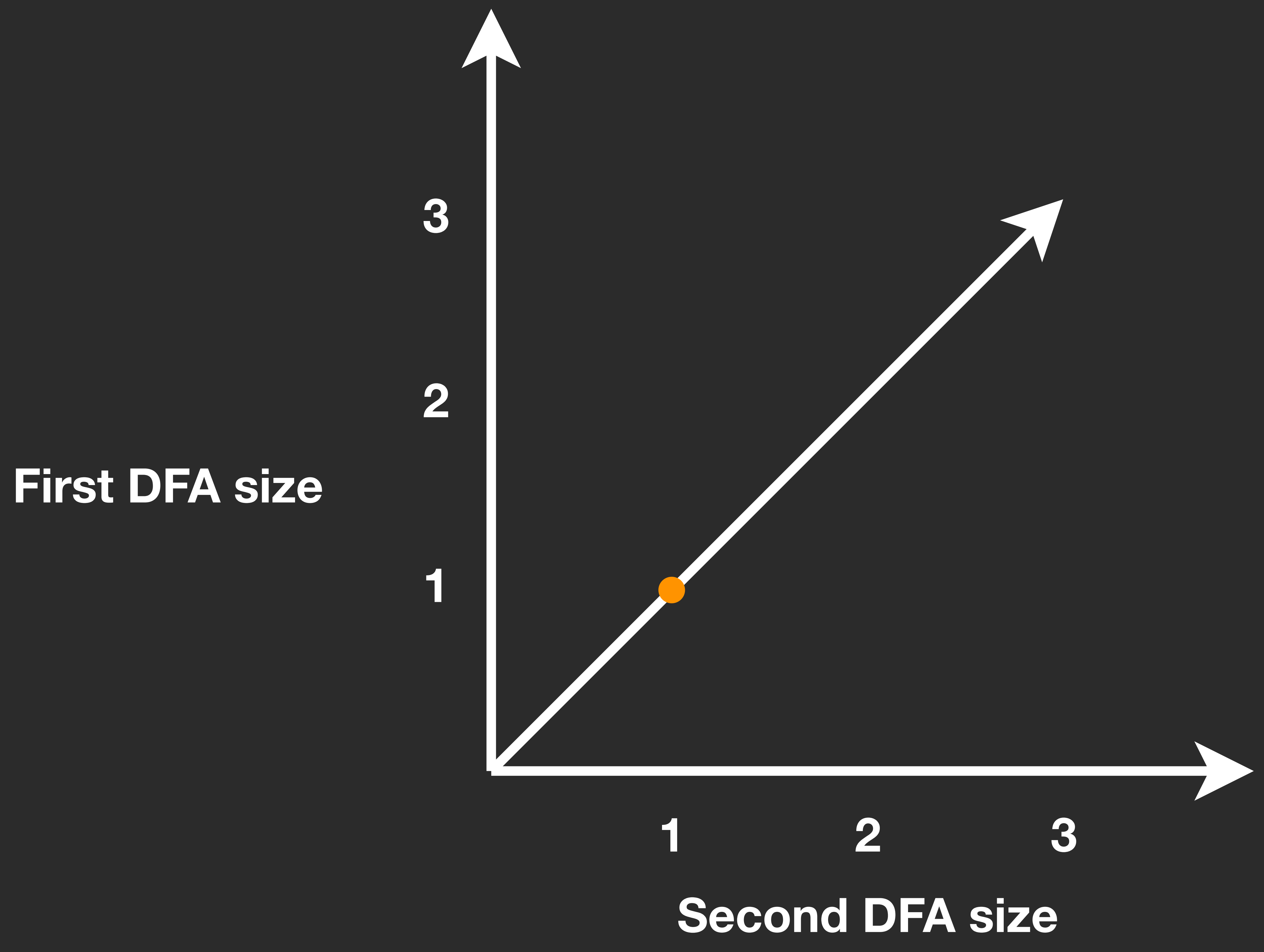
Finding the Pareto front of minimality



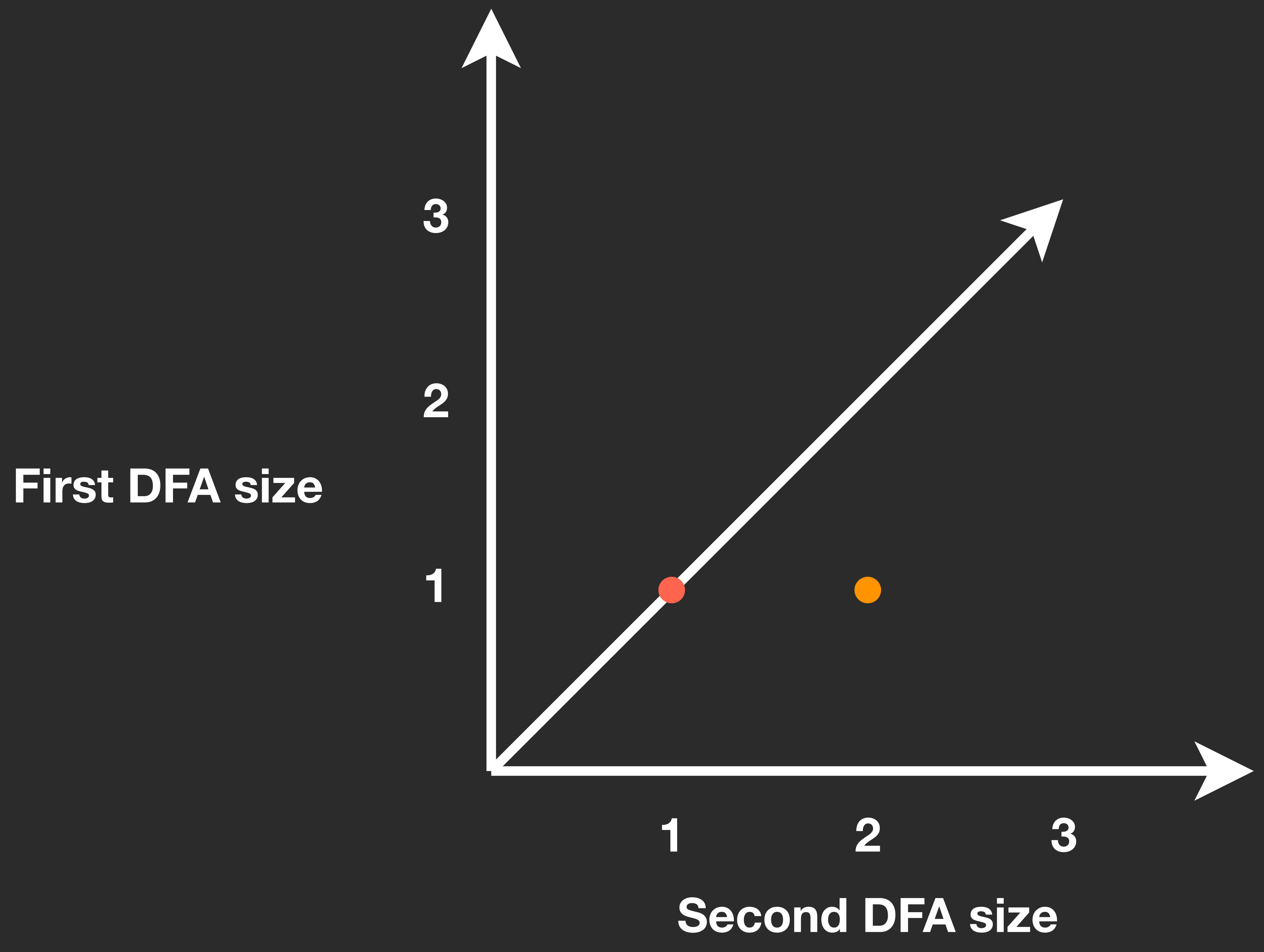
Finding the Pareto front of minimality



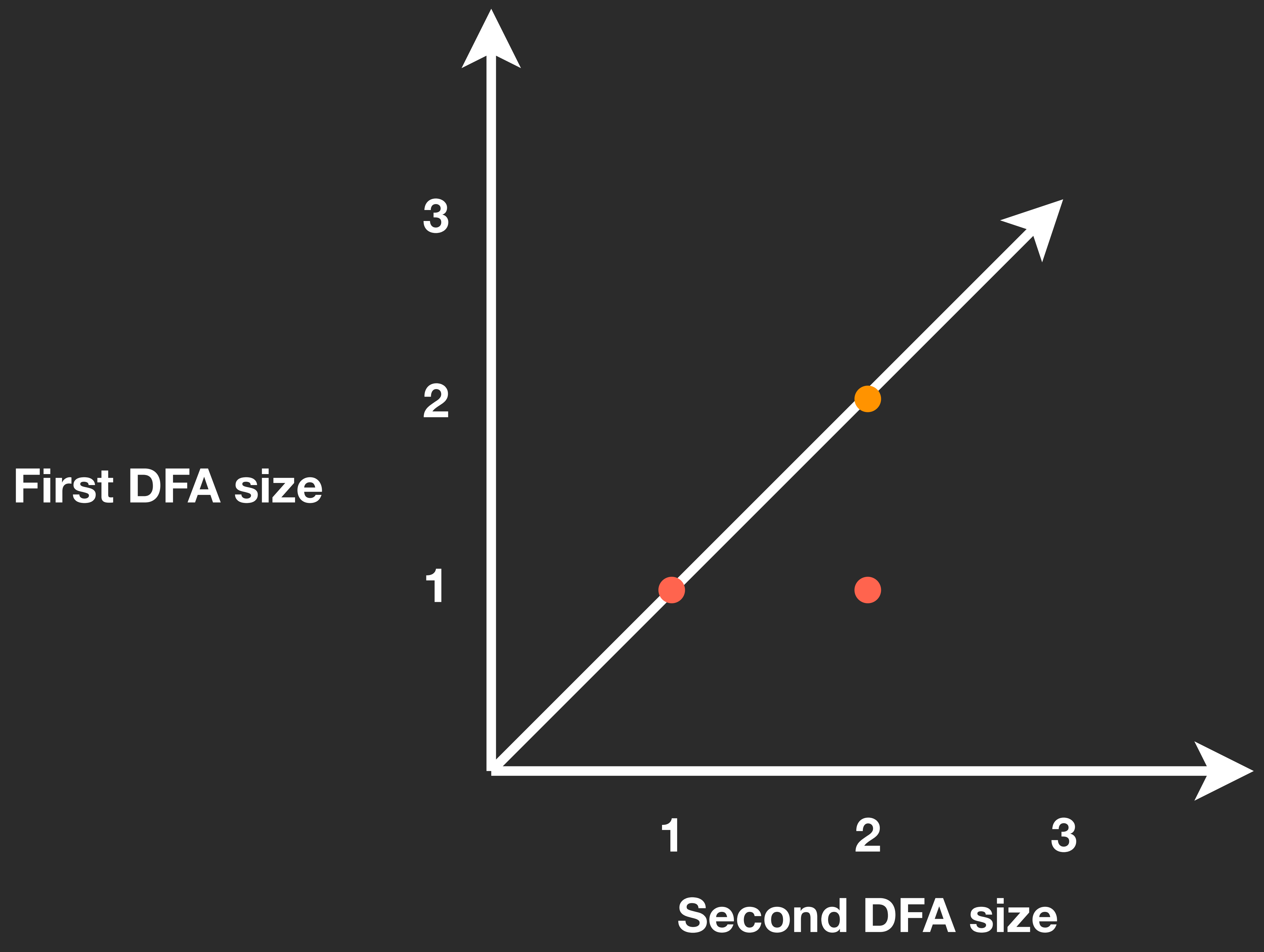
Finding the Pareto front of minimality



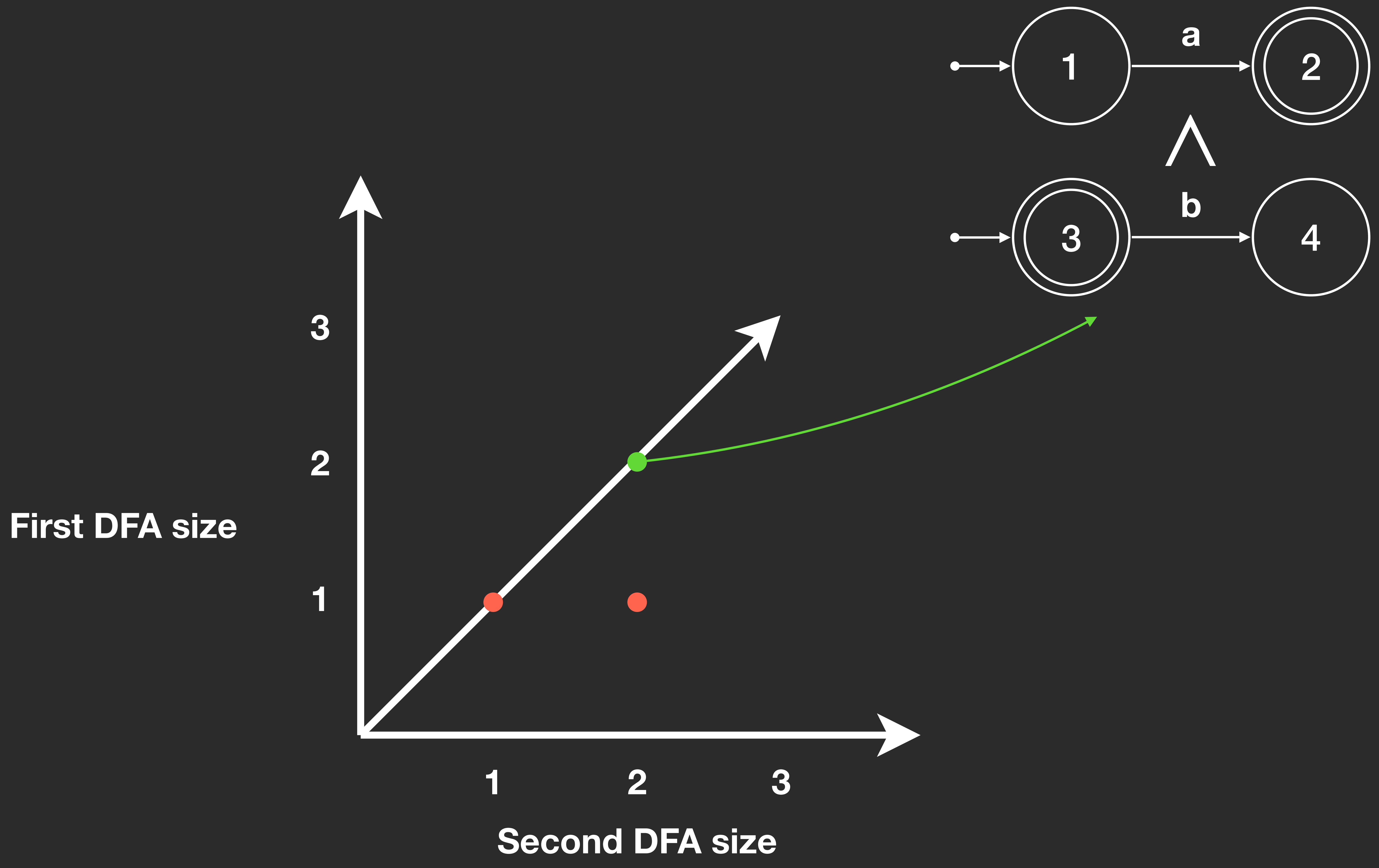
Finding the Pareto front of minimality



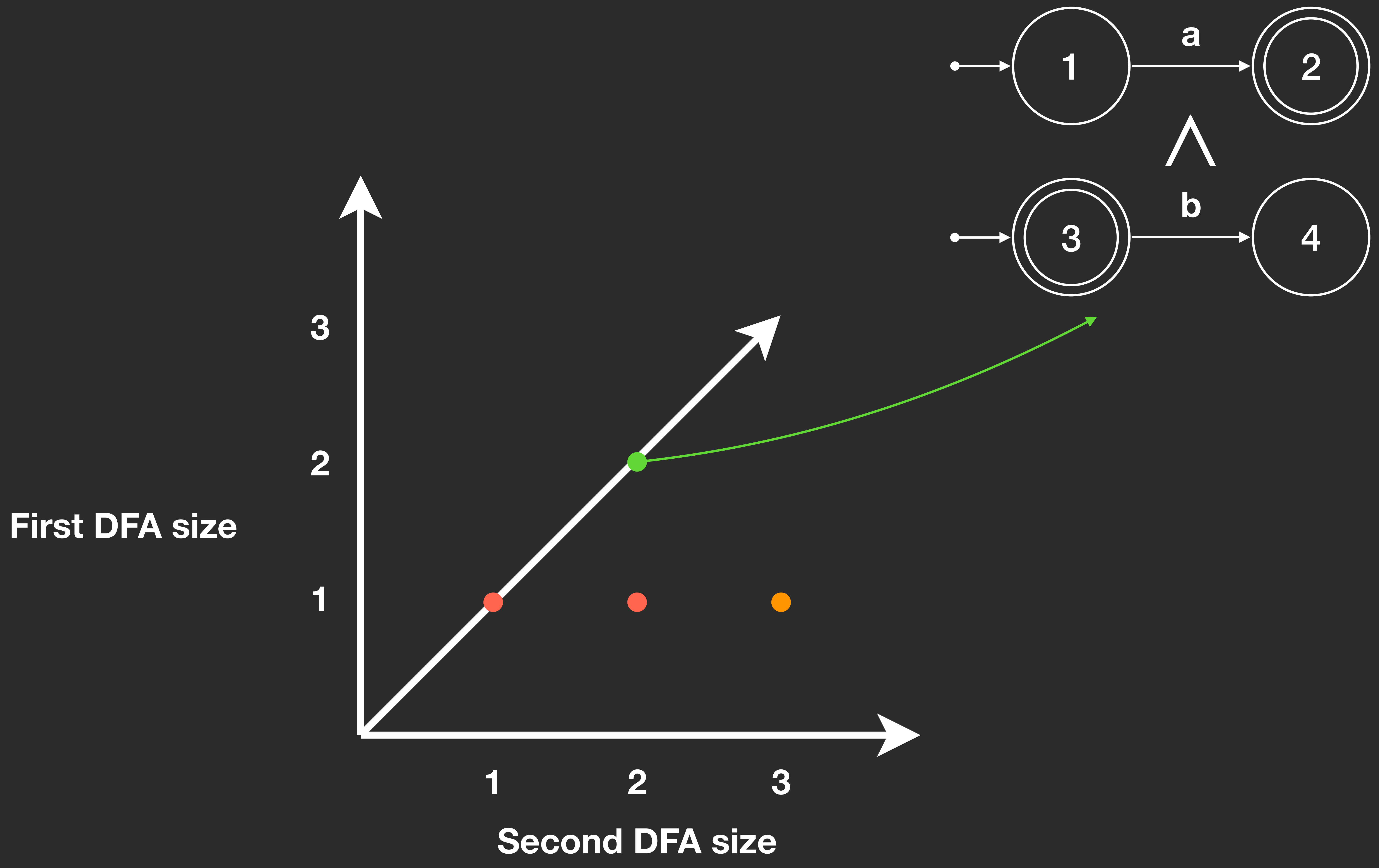
Finding the Pareto front of minimality



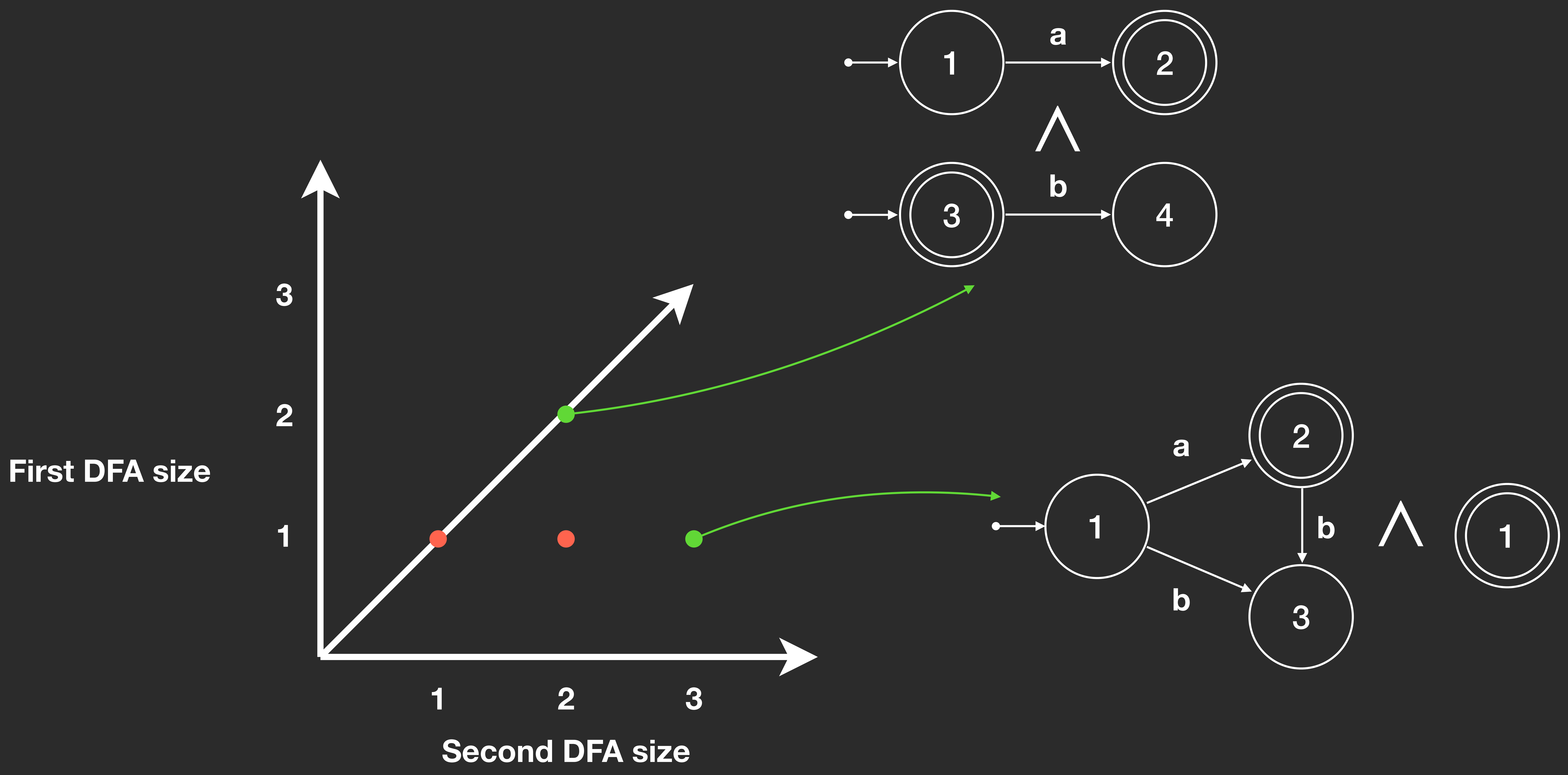
Finding the Pareto front of minimality



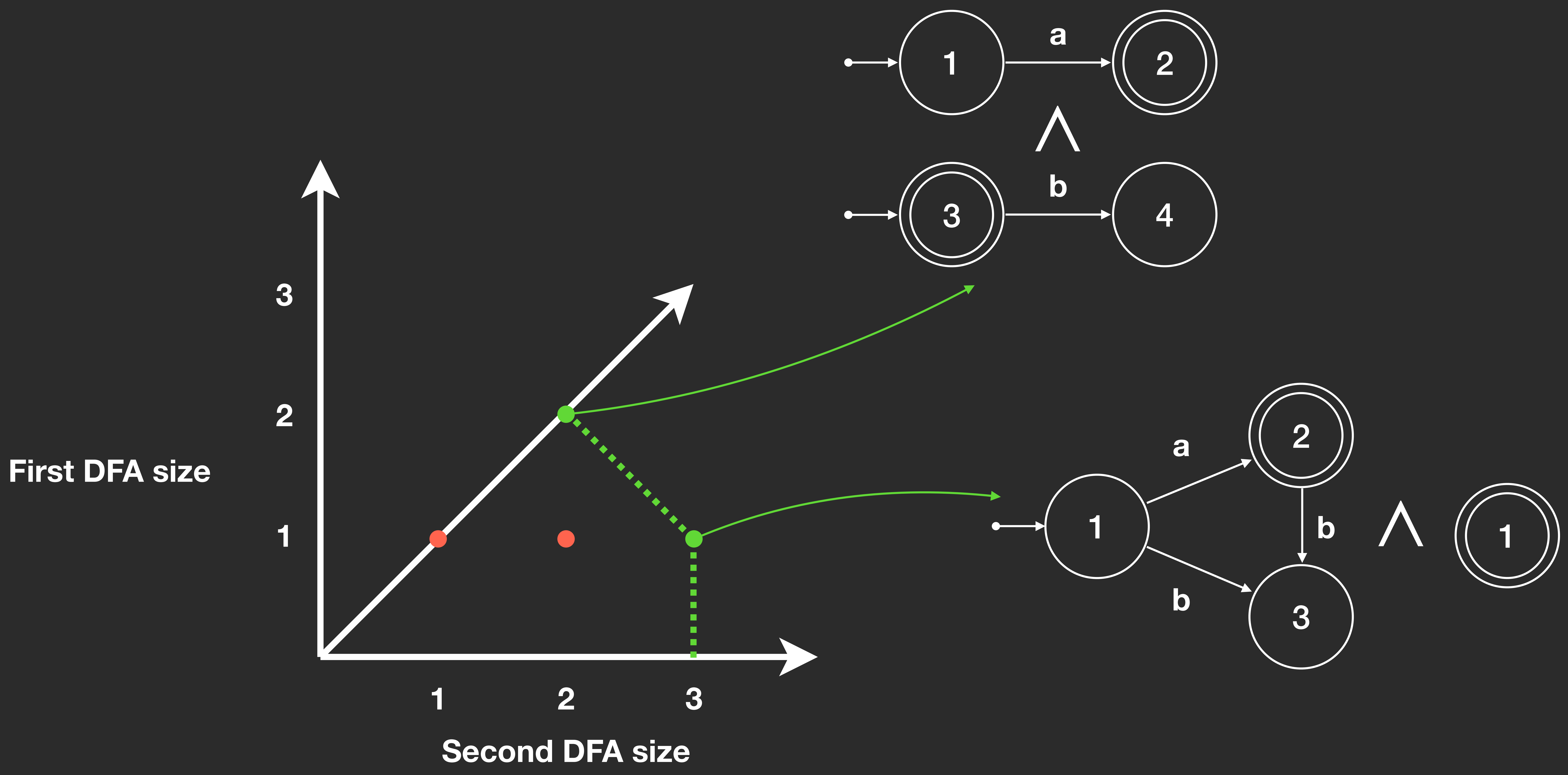
Finding the Pareto front of minimality



Finding the Pareto front of minimality



Finding the Pareto front of minimality



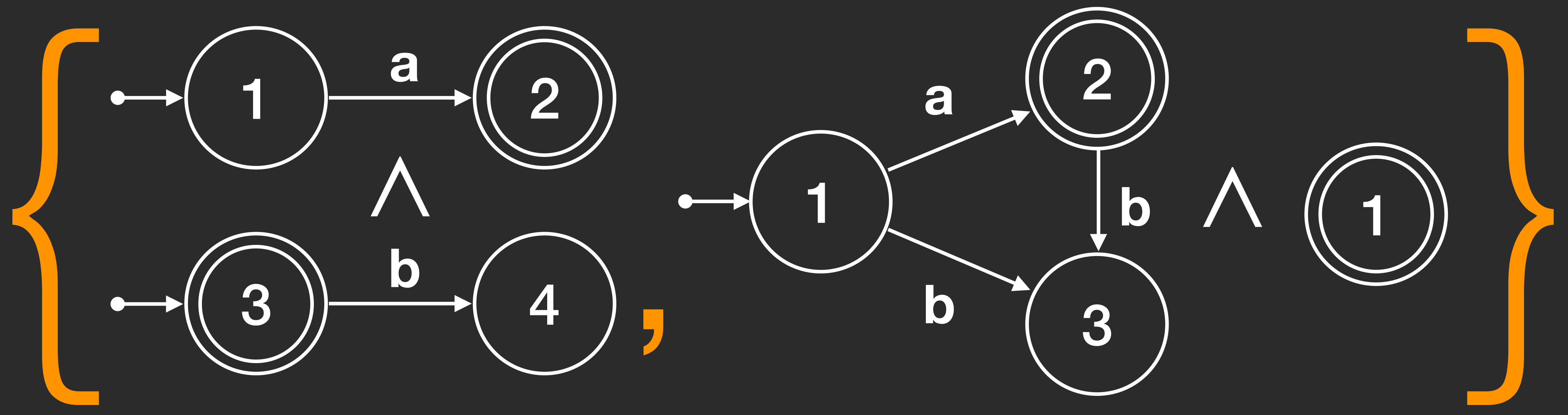
Structure of the talk

1. Technical details

Examples

Positive	**Negative**
a	Ø
	b
	ba
	ab
	aab

Our algorithm →



Pareto front of DFA decompositions

2. Scalability analysis

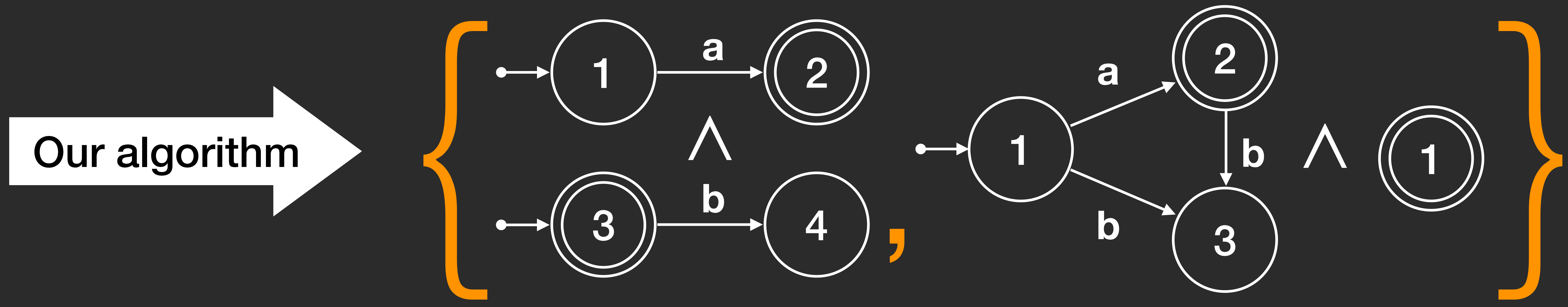
3. Learning from demonstrations

Structure of the talk

1. Technical details

Examples

Positive	Negative
a	ø
	b
	ba
	ab
	aab

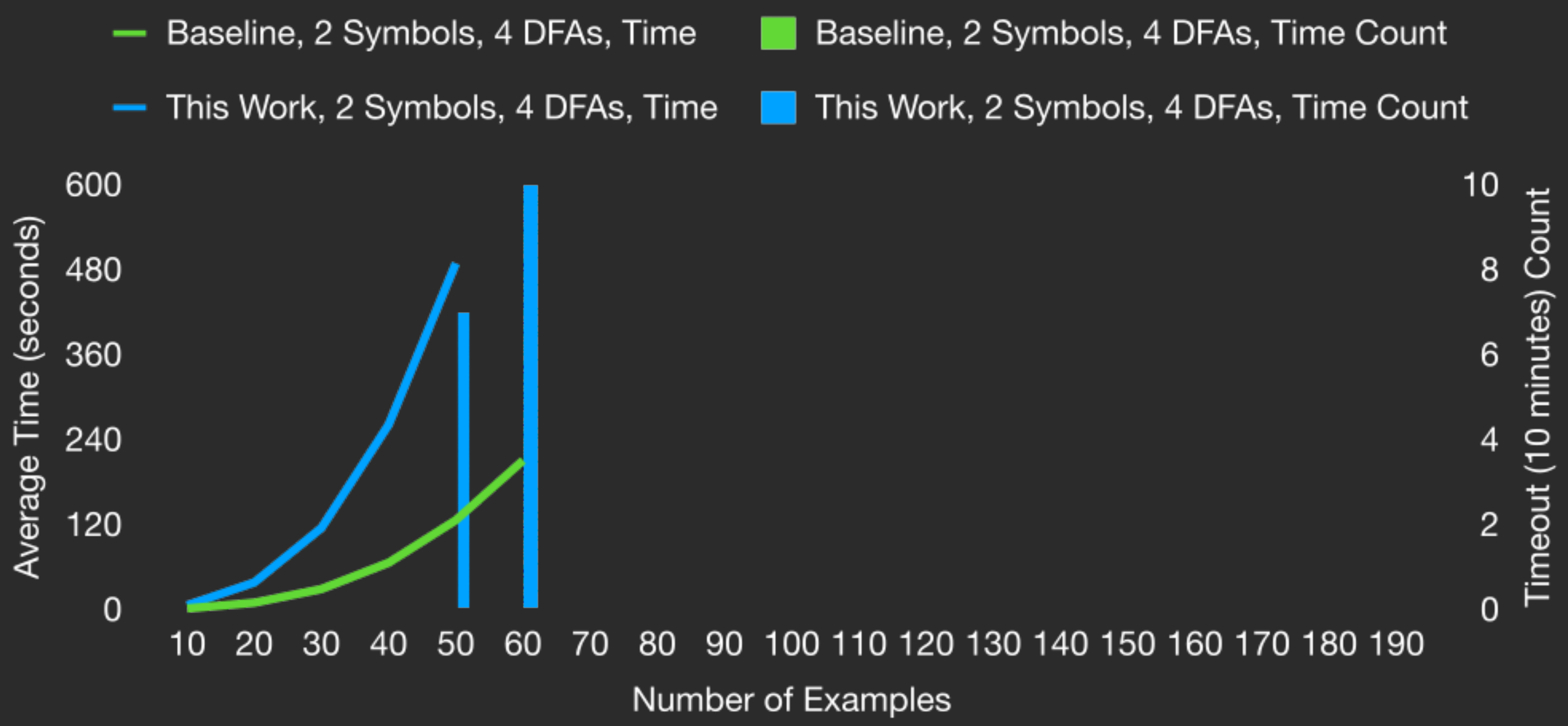


Pareto front of DFA decompositions

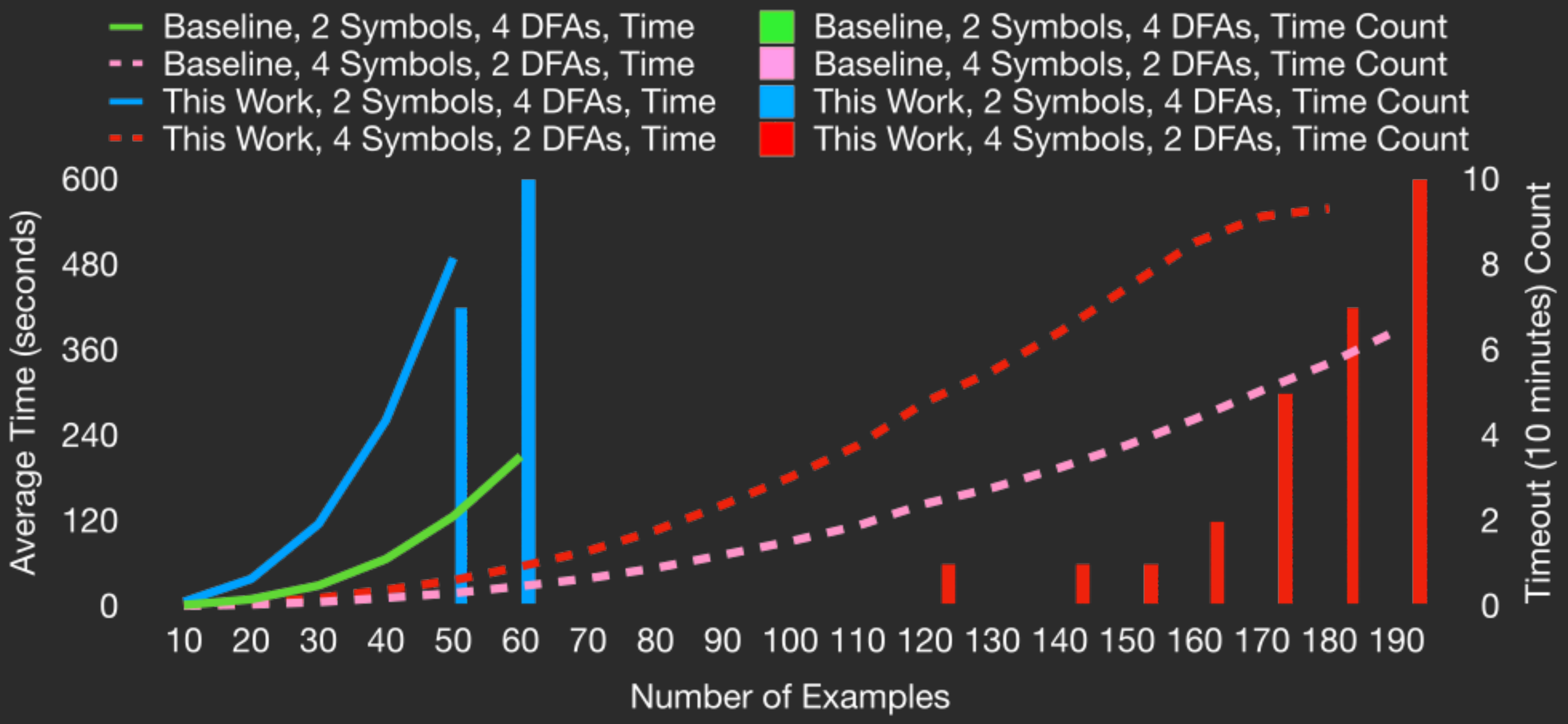
2. Scalability analysis

3. Learning from demonstrations

Overhead comparable to the monolithic baseline



Overhead comparable to the monolithic baseline



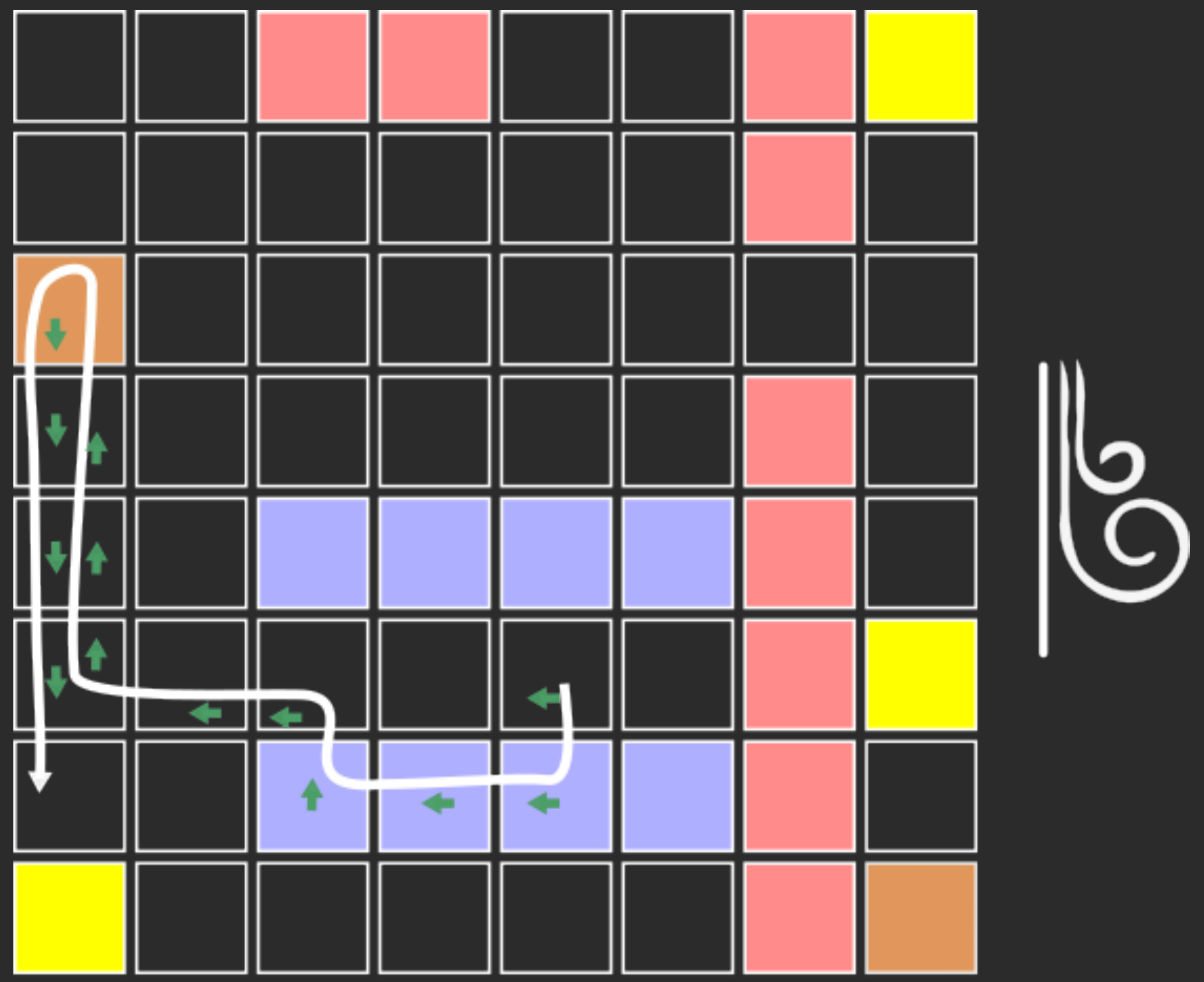
Structure of the talk

1. Technical details

2. Scalability analysis

3. Learning from demonstrations

Learning from demonstrations



Actions = {↑ ↓ ← →}

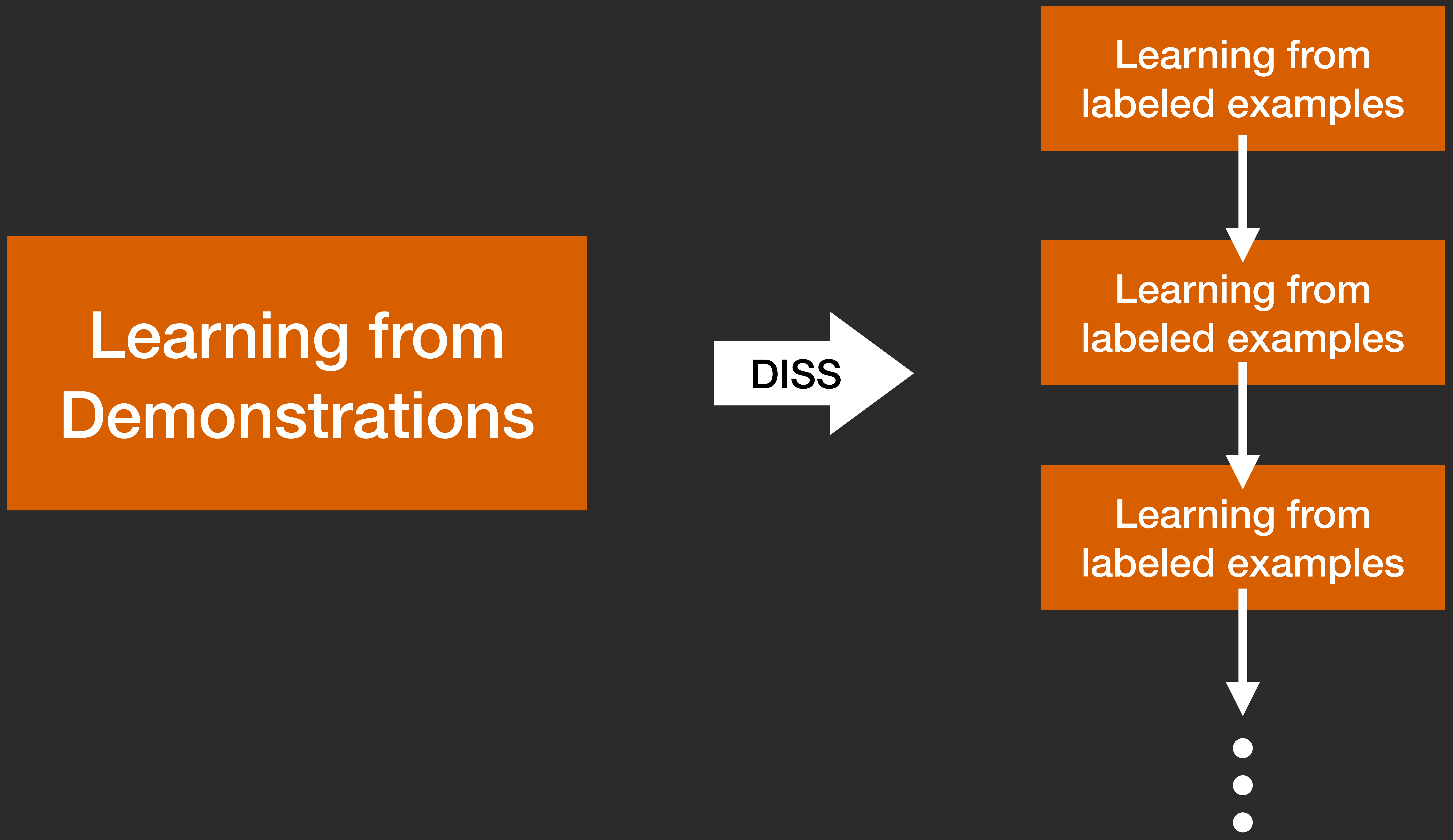
Pr(slip ↓) = 1/32

Demonstration Informed Specification Search (DISS)

Learning from Demonstrations

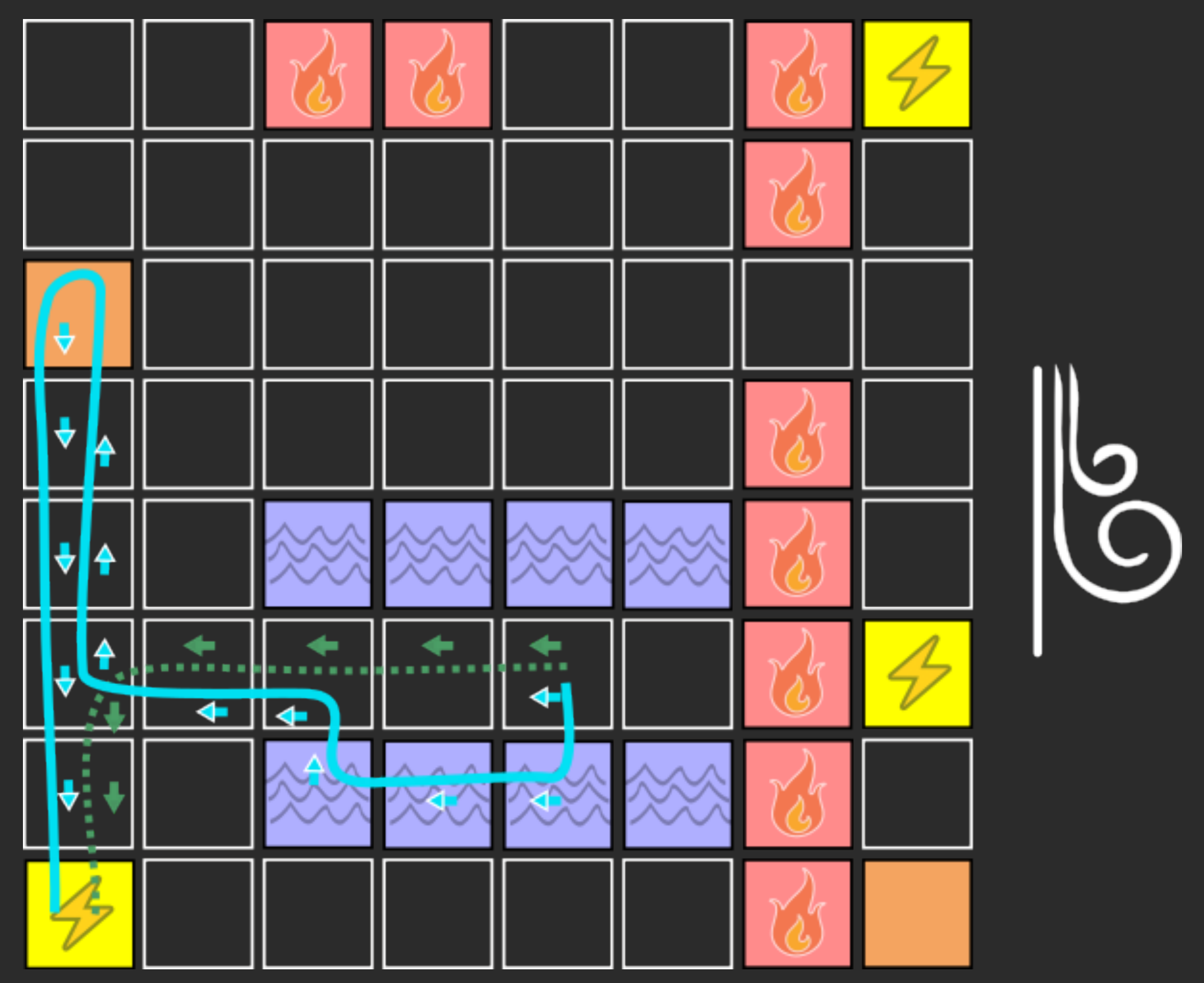
*Vazquez-Chanlatte, Marcell Jose. Specifications from Demonstrations: Learning, Teaching, and Control. Diss. UC Berkeley, 2022.

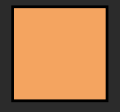
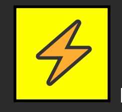
Demonstration Informed Specification Search (DISS)



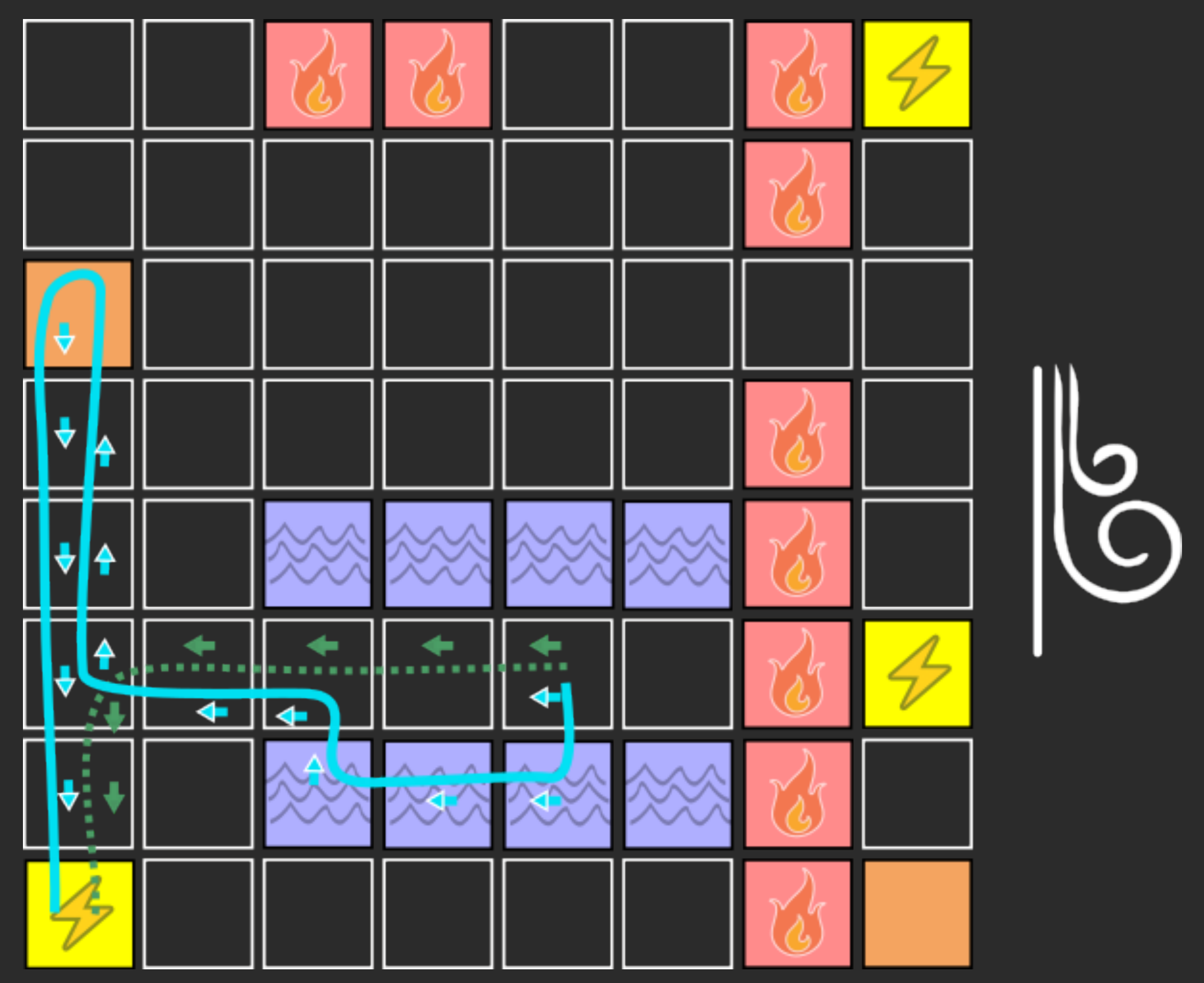
*Vazquez-Chanlatte, Marcell Jose. Specifications from Demonstrations: Learning, Teaching, and Control. Diss. UC Berkeley, 2022.

A helpful inductive bias from decompositions

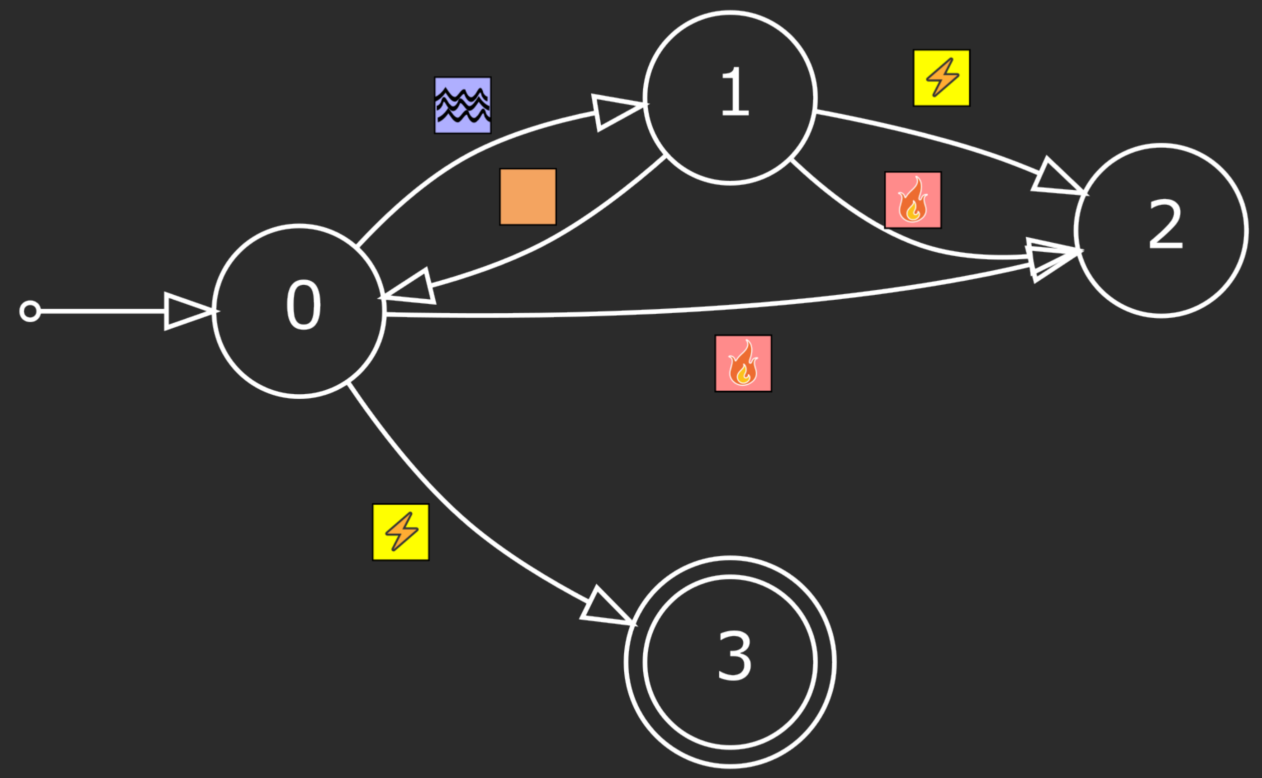


Reach  while avoiding . If you ever touch , you must then touch  before reaching .

A helpful inductive bias from decompositions

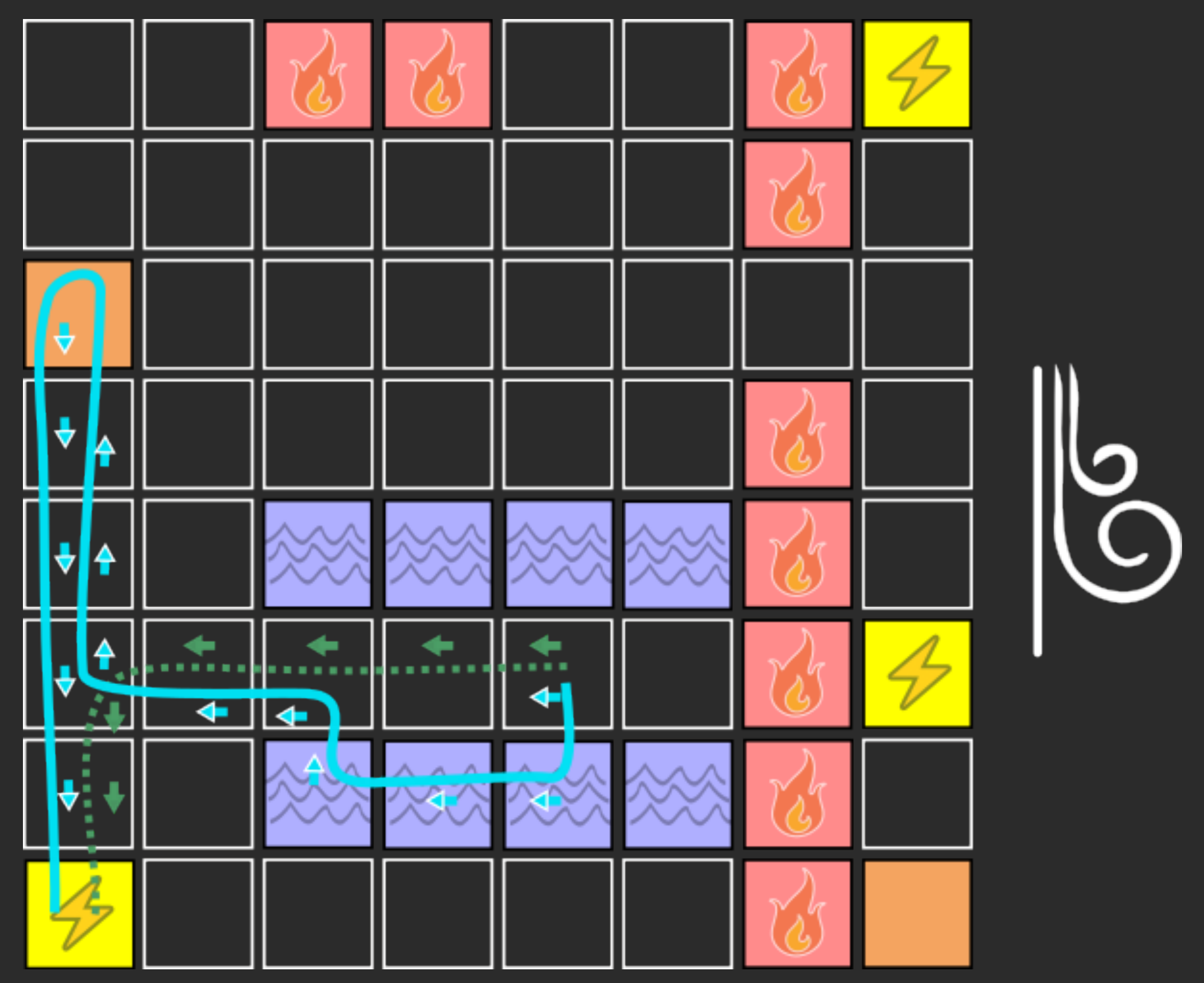


Reach ⚡ while avoiding 🔥. If you ever touch 🌊, you must then touch 🟧 before reaching ⚡.

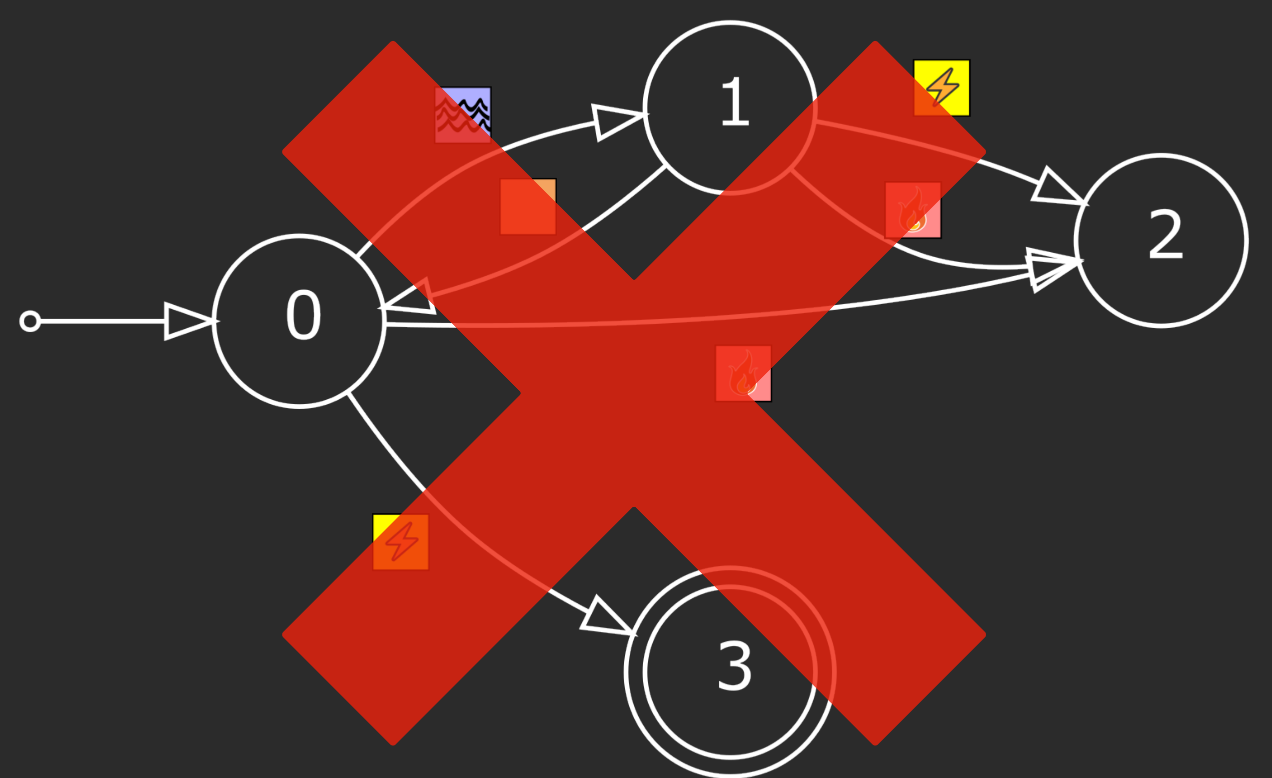


Identified monolithic DFA
(incorrect)

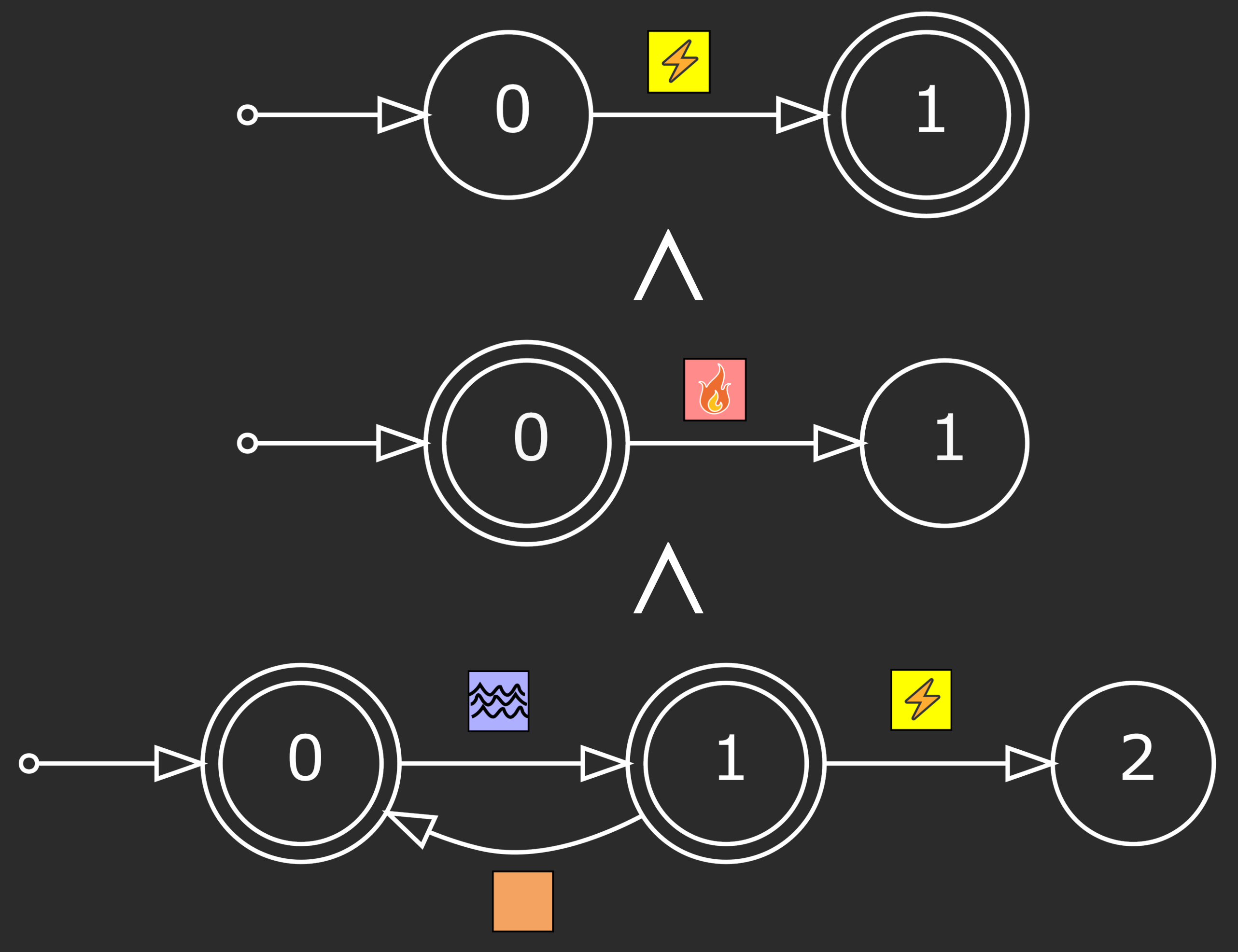
A helpful inductive bias from decompositions



Reach while avoiding . If you ever touch , you must then touch before reaching .

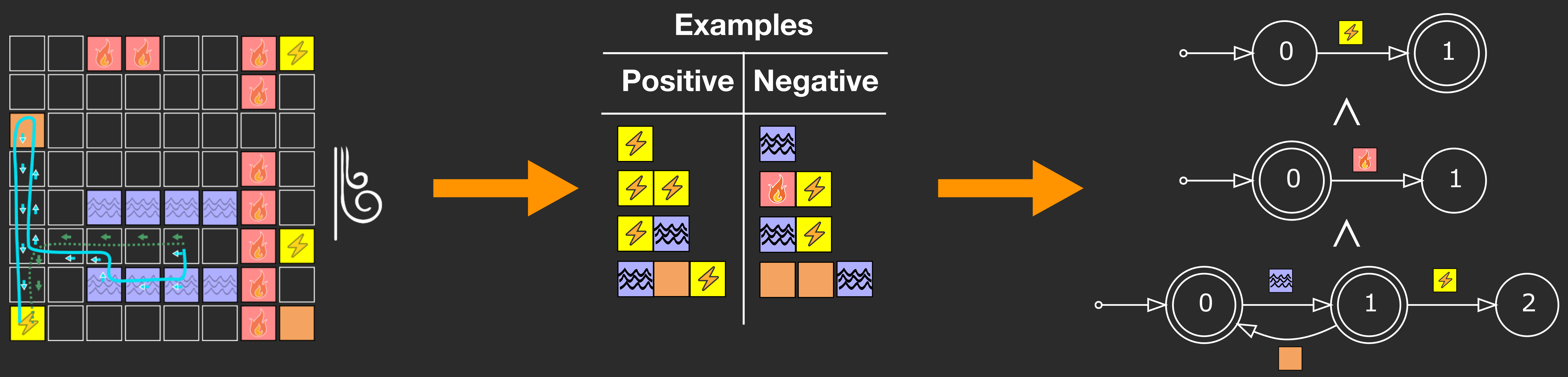


Identified monolithic DFA
(incorrect)



Identified DFA decomposition
(correct)

Conclusion



- Known symmetry-breaking optimization still missing from the encoding
- Easy to extend to disjunctions and boolean combinations of DFAs