# Describing Homing and Distinguishing Sequences for Nondeterministic Finite State Machines via Synchronizing Automata

Natalia Kushik and Nina Yevtushenko

Tomsk State University, Russia

# Motivation

Relies on pure mathematical interest ☺

- Finite automata and (synchronizing) experiments with them are well studied

However…

- Many reactive systems are often described using Finite State Machines (FSMs)
- Experiments with FSMs should be also considered

⇓

*We reduce the problem of deriving preset homing/distinguishing experiments for nondeterministic FSMs to solving the synchronizing problem for automata*

# Outline

1) Finite State Machines (FSMs) and Automata

2) Experiments with FSMs

3) Reducing the problem of deriving homing sequences for FSMs to deriving synchronizing sequences for automata

4) Reducing the problem of deriving distinguishing sequences for FSMs to deriving synchronizing sequences for automata

5) Complexity issues for distinguishing sequences for nondeterministic FSMs

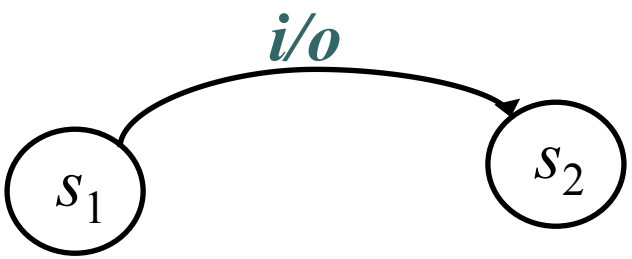
6) Conclusions and Future Work

# Finite State Machines and Automata

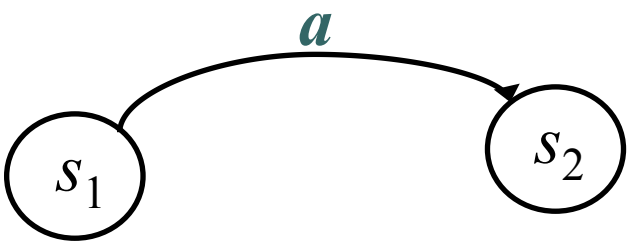
- *Finite state machines* (*FSMs*) and *Automata* describe the behavior of discrete event systems
- Differently from automata, FSMs usually model the behavior of reactive systems
- Reactive systems mostly work in query/request mode

⇓

- FSM transitions are labeled with input/output pairs




FSM transition

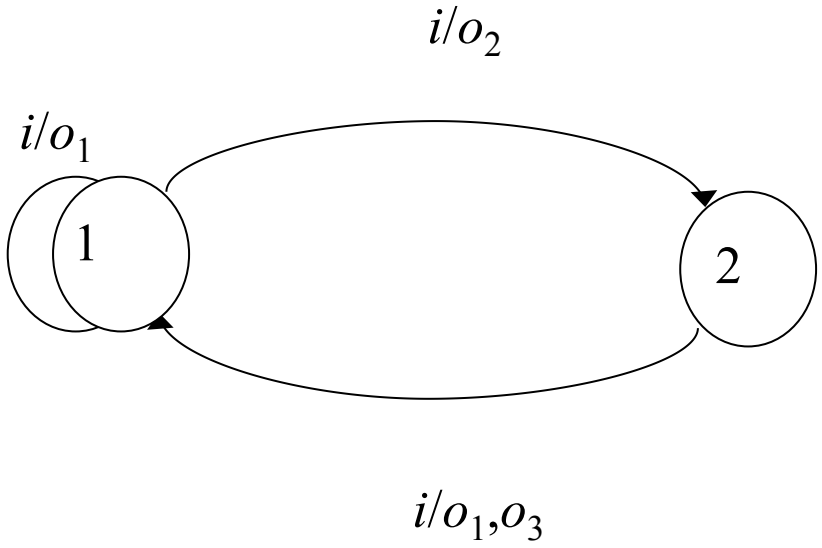



Automata transition

# Finite State Machine (FSM)

$\mathcal{S} = (S, I, O, h_S)$ is FSM

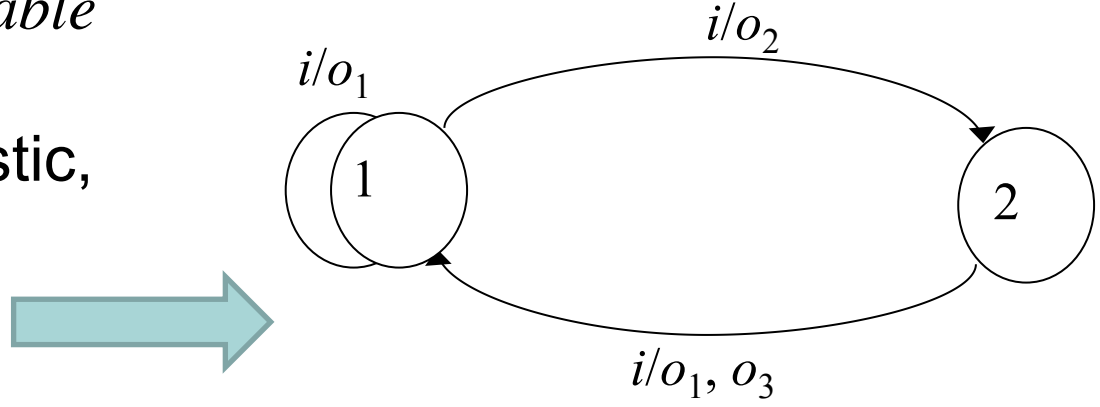
- $S$ is a finite nonempty set of states
- $I$ and $O$ are finite input and output alphabets
- $h_S \subseteq S \times I \times O \times S$ is the behavior relation




# FSM $\mathcal{S} = (S, I, O, h_S)$ can be

- *deterministic* if for each pair $(s, i) \in S \times I$ there exists at most one pair $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$
  otherwise, $\mathcal{S}$ is *nondeterministic*
- *complete* if for each pair $(s, i) \in S \times I$ there exists $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$
  otherwise, $\mathcal{S}$ is *partial*
- *observable* if for each triple $(s, i, o) \in S \times I \times O$ there exists at most one state $s' \in S$ such that $(s, i, o, s') \in h_S$
  otherwise, $\mathcal{S}$ is *nonobservable*

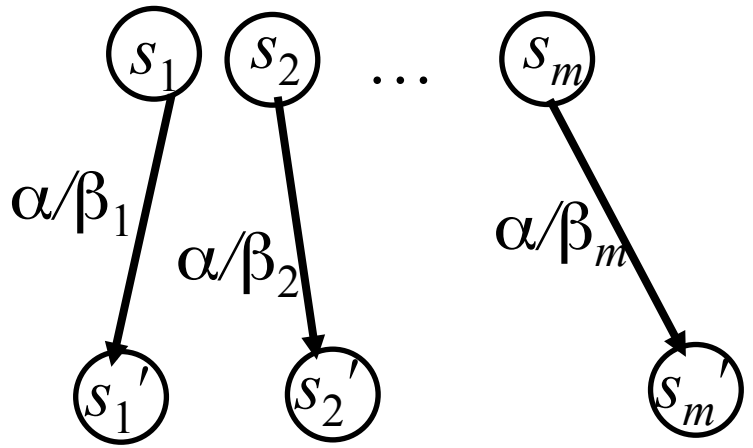
This FSM is nondeterministic, complete and observable



# Distinguishing sequence

- *Distinguishing* = *separating* for nondeterministic machines
- A distinguishing (input) sequence $\alpha$ allows to determine the initial state of the machine under experiment
- After applying $\alpha$ at any state $s$ and observing an output response $\beta$ the initial state $s$ becomes known

Separating sequence $\alpha$



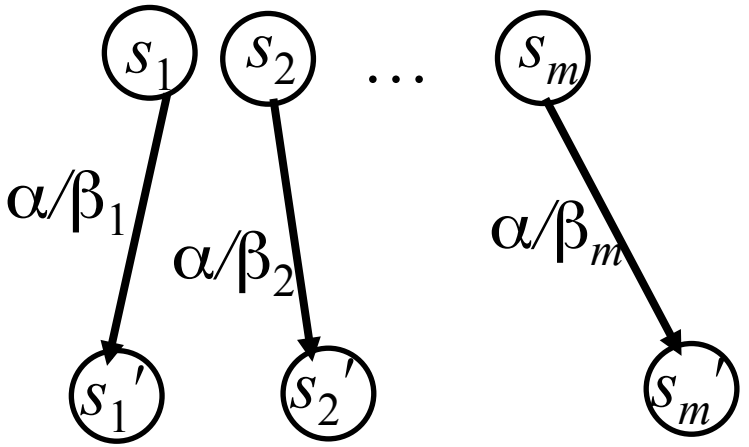

$out(s_i, \alpha) \cap out(s_j, \alpha) = \varnothing$

(Preset) distinguishing experiment = applying $\alpha$ + observing $\beta_i$ + drawing a conclusion about $s_i$

# Homing sequence

- A homing (input) sequence $\alpha$ allows to determine the final state of the machine under experiment after applying $\alpha$
- After applying $\alpha$ at any state $s$ and observing an output response $\beta$ the final state $s'$ becomes known

Homing sequence $\alpha$

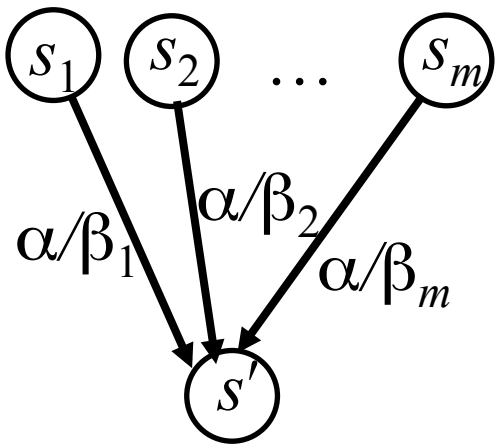



(Preset) homing experiment = applying $\alpha$ + observing $\beta_i$ + drawing a conclusion about $s_i'$

# Synchronizing sequence

- A synchronizing sequence $\alpha$ takes the machine under experiment to a given state after applying $\alpha$
- After applying $\alpha$ at any state the final state is $s'$

Synchronizing sequence $\alpha$




For a synchronizing sequence output reactions are not taken into account

$\Rightarrow$ synchronizing sequences are usually derived for automata (machines without outputs)

# Does there exist a distinguishing sequence?

*The decision problem is considered*

*DISTINGUISHING problem*
**Input:** complete deterministic FSM $\mathcal{S} = (S, I, O, h_S)$
**Output:** Does there exist a distinguishing sequence for $\mathcal{S}$?

The problem of checking the existence of a distinguishing sequence for **deterministic** FSMs is PSPACE-complete

Lee, D., Yannakakis, M., 1994

# One way to derive a distinguishing sequence for nondeterministic FSM

Derive a truncated successor tree (TST)

$\exists\, o$ $((s_1, i_j, o, s_1', ) \in h_S$ &

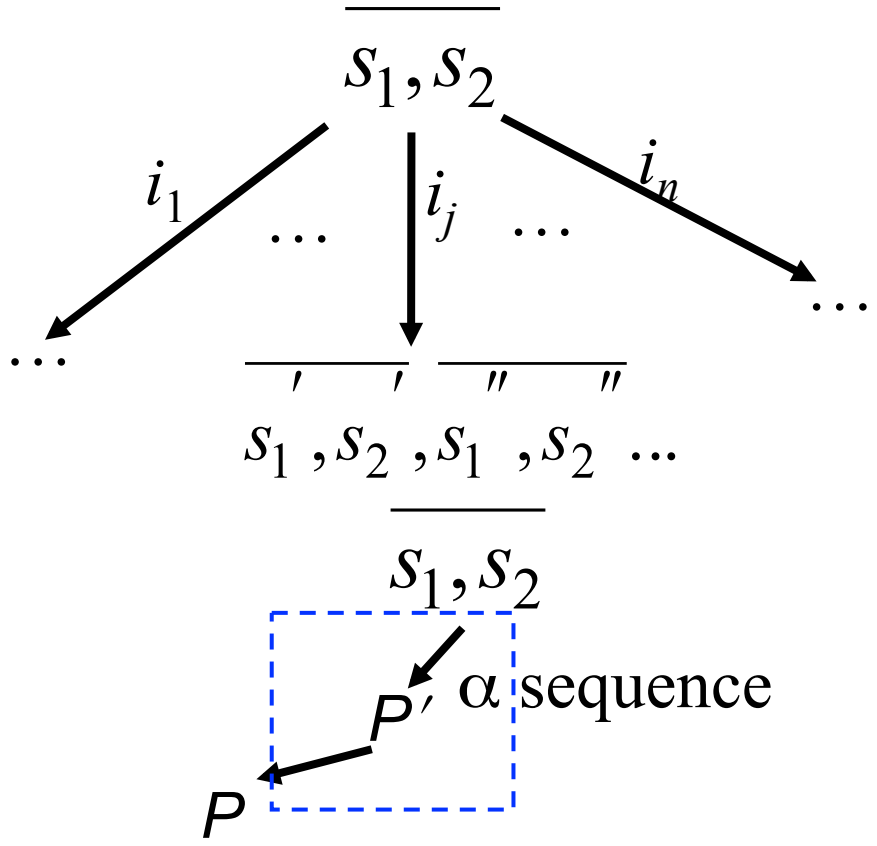
$(s_2, i_j, o, s_2') \in h_S)$

- Truncating rules

**Rule 1** $P$ is the empty set

**Rule 2** Set $P$ contains a subset that labels another node of the path from the root to the node labeled by the set $P$

**Rule 3** $P$ contains a singleton




**α is a distinguishing sequence iff it labels the path truncated by Rule 1**

# One way to derive a homing sequence for nondeterministic FSM

Derive a truncated successor tree (TST)

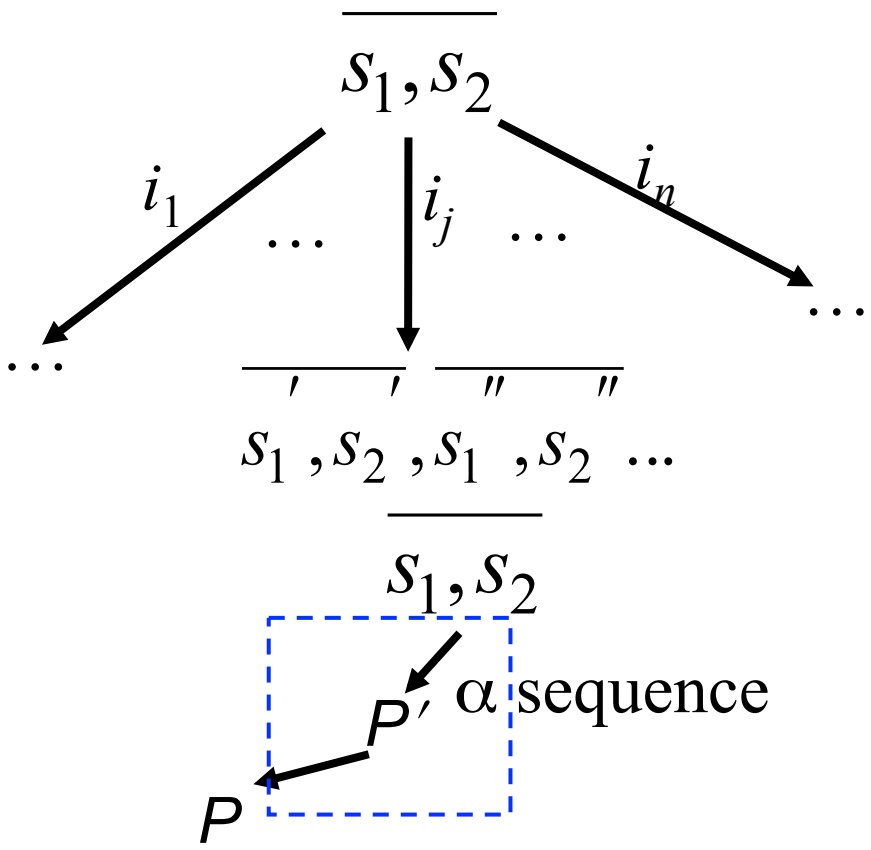
$\exists\, o\ ((s_1, i_j, o, s_1') \in h_S\ \&\ (s_2, i_j, o, s_2') \in h_S)$

- Truncating rules

**Rule 1** *P* is the empty set

**Rule 2** Set *P* apart from singletons contains a set labeling a node at a higher tree level

**Rule 3** *P* contains only singletons




**α is a homing sequence iff it labels the path truncated by Rule 1 or Rule 3**

# Another way to derive homing and distinguishing sequences

Let's derive homing/distinguishing sequences for nondeterministic FSMs without addressing truncated trees

Huge truncated successor tree

Compact automaton that preserves all the necessary sequences

Why and what it gives to us?

- Always nice to have an alternative method
- Might help to estimate the complexity of related decision and derivation problems for distinguishing and homing sequences
- Can help to construct special FSM classes with low complexity bounds

# Idea… Let's look over the languages

Given complete nondeterministic FSM $\mathcal{S} = (S, I, O, h_S)$

$L_{home}(\mathcal{S})$ is the set of all homing sequences of $\mathcal{S}$

$L_{dist}(\mathcal{S})$ is the set of all distinguishing sequences of $\mathcal{S}$

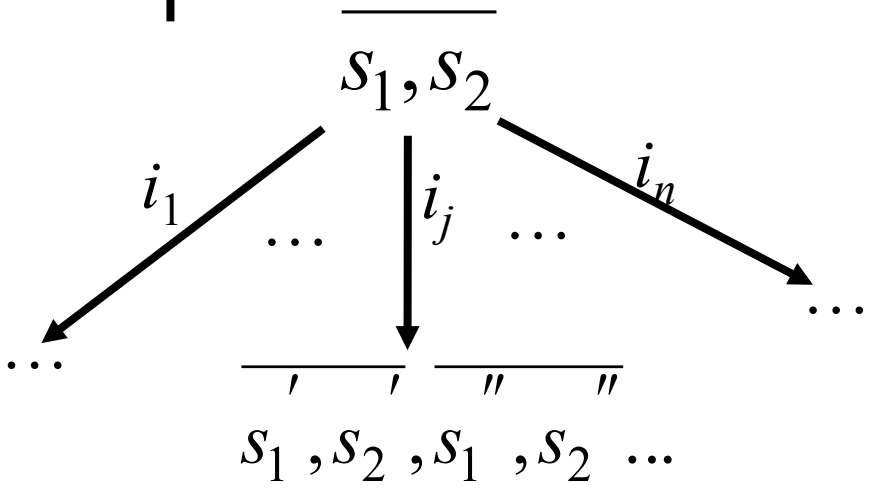
Our objective : to derive an automaton A with the set $L_{synch}(A)$ of synchronizing sequences such that

- $L_{dist}(\mathcal{S}) = L_{synch}(A)$
- $L_{home}(\mathcal{S}) = L_{synch}(A)$

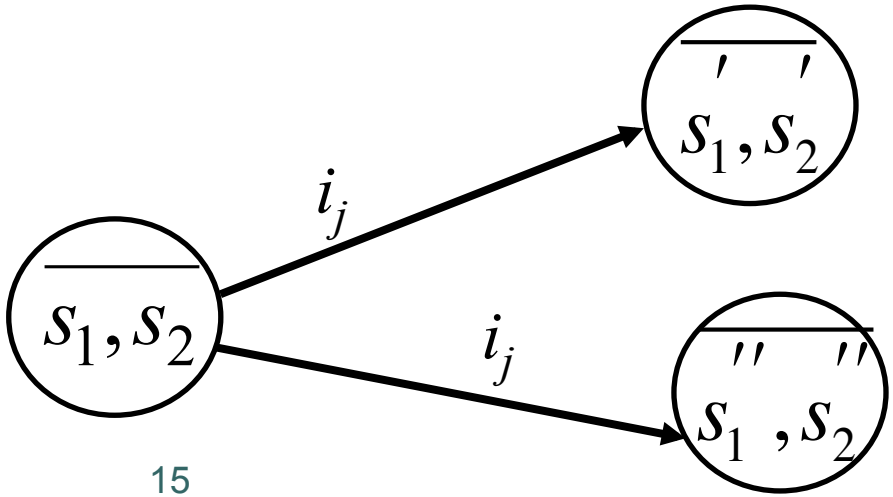
$\Downarrow$

*And the question is : does there exist such automaton and if it exists then how to derive such automaton?*

# Deriving automata of interest




The truncated successor tree looks like this




These are the automaton transitions

The designated *sink* state where each path of interest is terminated

# Deriving an automaton $S^2_{home}$

$S^2_{home}$ states : $\overline{s_j, s_k}$ , $j < k$,
designated state *sink*

$S^2_{home}$ actions : inputs of FSM $S$

**For** each input $i \in I$

**For** each state $\overline{s_j, s_k}$ of the automaton $S^2_{home}$

Add to the automaton $S^2_{home}$ the transition ( $\overline{s_j, s_k}$, $i$, $\overline{s_p, s_t}$ ), if $\overline{s_p, s_t}$ is the *io*-successor of $\overline{s_j, s_k}$ for some output $o \in O$

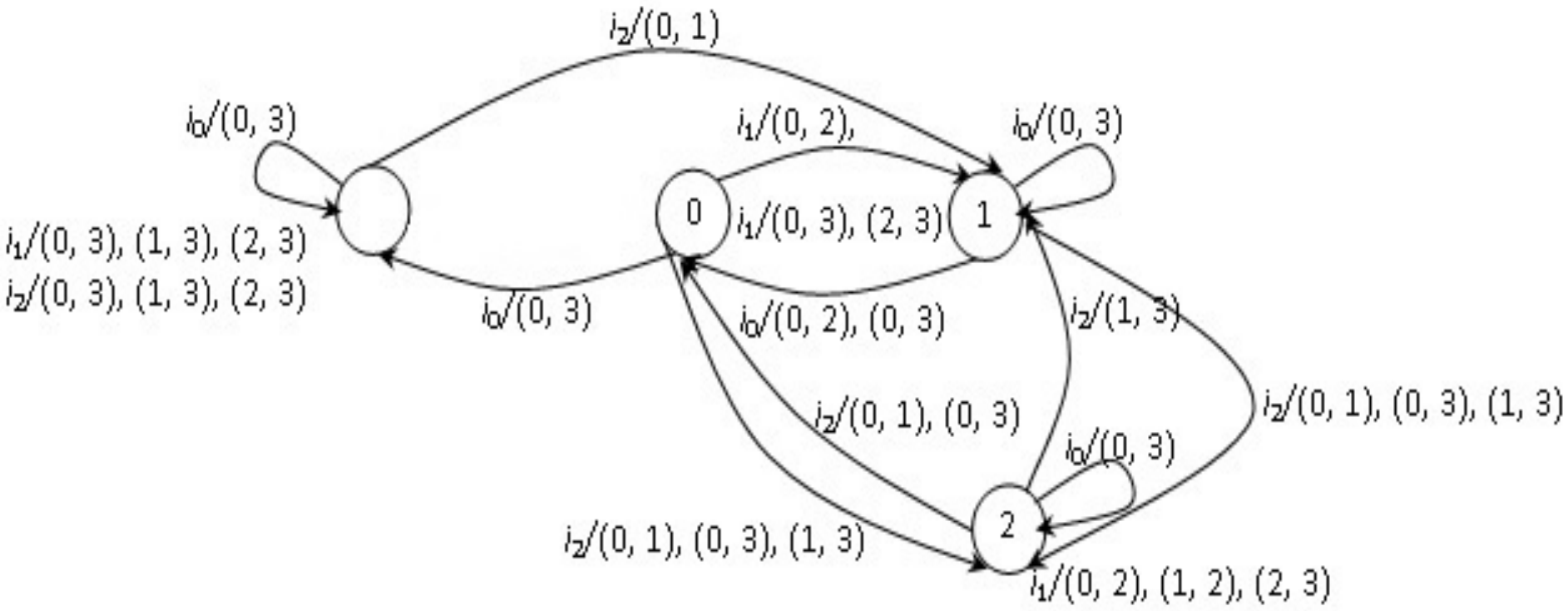
Add to the automaton $S^2_{home}$ the transition ( $\overline{s_j, s_k}$ , $i$, *sink*) if for each output $o \in O$ the *io*-successor of $\overline{s_j, s_k}$ is a singleton or states $s_j$ and $s_k$ are separated by the input $i$

Add to the automaton $S^2_{home}$ the transition (*sink*, $i$, *sink*)

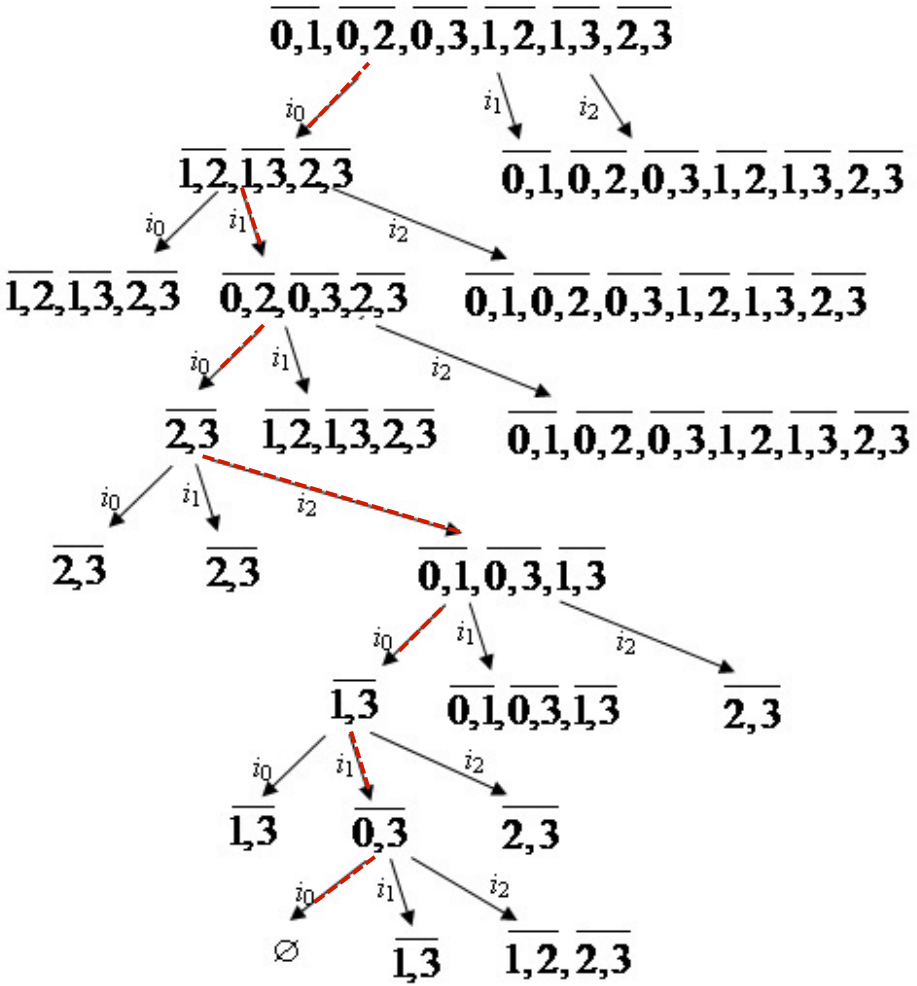
**EndFor**

**EndFor**

# Example… FSM $\mathcal{S}_4$



# $S_4$ truncated successor tree

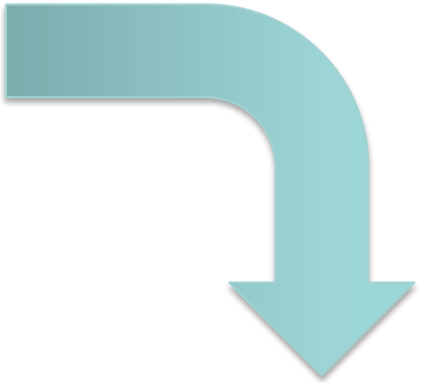



A shortest homing sequence traverses a sequence of all state subsets which are not singletons

$\overline{0,1,2,3}$, $\overline{1,2,3}$, $\overline{0,2,3}$, $\overline{2,3}$, $\overline{0,1,3}$, $\overline{1,3}$, $\overline{0,3}$

# FSM $\mathcal{S}_4$ and its automaton $S^2_{home}$

| Input/State | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $i_0$ | $3/(0,3)$ | $1/(0,3)$ | $2/(0,3)$ | $3/(0,3)$ |
| $i_1$ | $1/(0,2)$<br>$1/(0,3)$<br>$1/(2,3)$ | $0/(0,2)$<br>$0/(0,3)$ | $2/(0,2)$<br>$2/(1,2)$<br>$2/(2,3)$ | $3/(0,3)$<br>$3/(1,3)$<br>$3/(2,3)$ |
| $i_2$ | $2/(0,1)$<br>$2/(0,3)$<br>$2/(1,3)$ | $2/(0,1)$<br>$2/(0,3)$<br>$2/(1,3)$ | $0/(0,1)$<br>$0/(0,3)$<br>$1/(1,3)$ | $3/(0,3)$<br>$3/(1,3)$<br>$3/(2,3)$<br>$1/(0,1)$ |



| Input/State | $\overline{0,1}$ | $\overline{0,2}$ | $\overline{0,3}$ | $\overline{1,2}$ | $\overline{1,3}$ | $\overline{2,3}$ | *sink* |
|---|---|---|---|---|---|---|---|
| $i_0$ | $\overline{1,3}$ | $\overline{2,3}$ | *sink* | $\overline{1,2}$ | $\overline{1,3}$ | $\overline{2,3}$ | *sink* |
| $i_1$ | $\overline{0,1}$ | $\overline{1,2}$ | $\overline{1,3}$ | $\overline{0,2}$ | $\overline{0,3}$ | $\overline{2,3}$ | *sink* |
| $i_2$ | *sink* | $\overline{0,2}$<br>$\overline{1,2}$ | $\overline{2,3}$ | $\overline{0,2}$<br>$\overline{1,2}$ | $\overline{1,2}$<br>$\overline{2,3}$ | $\overline{0,1}$<br>$\overline{0,3}$<br>$\overline{1,3}$ | *sink* |

A shortest synchronizing sequence for $S^2_{home}$ is $i_0 i_1 i_0 i_2 i_0 i_1 i_0$

# Deriving an automaton $S^2_{dist}$

$S^2_{dist}$ states : $\overline{s_j, s_k}$ , $j < k$,
designated state *sink*

$S^2_{dist}$ actions : inputs of FSM $S$

**For** each input $i \in I$

**For** each state $\overline{s_j, s_k}$ of the automaton $S^2_{dist}$

Add to the automaton $S^2_{dist}$ the transition $(\overline{s_j, s_k}, i, sink)$,
if states $s_j$ and $s_k$ are separated by the input $i$

Add to the automaton $S^2_{dist}$ the transition $(\overline{s_j, s_k}, i, \overline{s_p, s_t})$,
if for each $o \in O$, the $io$-successors of states $s_j$ and $s_k$ do not coincide and $\overline{s_p, s_t}$ is the $io'$-successor of $\overline{s_j, s_k}$ for some $o' \in O$

Add to the automaton $S^2_{dist}$ the transition $(sink, i, sink)$

**EndFor**

**EndFor**

# Properties of $S^2_{home}$ and $S^2_{dist}$

- $L_{home}(\mathcal{S}) = L_{synch}(S^2_{home})$
- $L_{dist}(\mathcal{S}) = L_{synch}(S^2_{dist})$
- FSM $\mathcal{S}$ is *homing* if and only if the automaton $S^2_{home}$ is *synchronizing*
- There exists a *distinguishing* sequence for $\mathcal{S}$ if and only if the automaton $S^2_{dist}$ is *synchronizing*
- $S^2_{home}$ can be nondeterministic
- $S^2_{dist}$ can be nondeterministic and partial

# Some complexity issues…

- The number of $S^2_{home}$ and $S^2_{dist}$ transitions does not exceed $| I | n^2 (n - 1)$,

  $|S| = n$

- When $| I |$ is $O(n^k)$, $S^2_{home}$ and $S^2_{dist}$ can be derived in a polynomial time and can be stored in a polynomial space

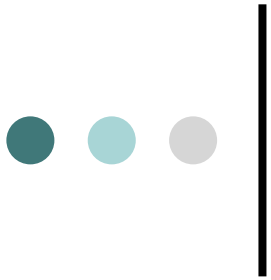
- The problem of checking the existence of a distinguishing sequence for complete nondeterministic observable FSMs is PSPACE-complete

# Conclusions and future work

- The problem of deriving homing/distinguishing sequences for a possibly nondeterministic FSM can be reduced to that of deriving a synchronizing sequence for a nondeterministic (partial) automaton

*It is interesting…*

- To study specific FSM classes that result in automata $S^2_{home}$ and $S^2_{dist}$ with synchronizing sequences of polynomial length
- To consider adaptive homing and distinguishing experiments and check whether the similar reduction is possible
- …



# Thank you!