Casting

Synchronizing Strategies under Partial Observability

Submitted

Kim G. Larsen   Simon Laursen   Jiří Srba

Aalborg University, Department of Computer Science, Denmark

May 22, 2014

SEVENTH FRAMEWORK PROGRAMME

Content

Synchronizing Words

- Given an automaton, can you find a word that brings the automaton to the same single state no-matter where it started?
- Seen practical applications in
  - biocomputing
  - model-based testing, and
  - robotics.
- The Černý conjecture is one of the longest standing conjectures in automata theory

Černý conjecture - 1964

The length of the shortest synchronizing word for any n-state DFA is at most $(n-1)^2$.

Synchronizing Words

- Given an automaton, can you find a word that brings the automaton to the same single state no-matter where it started?
- Seen practical applications in
  - biocomputing
  - model-based testing, and
  - robotics.
- The Černý conjecture is one of the longest standing conjectures in automata theory

Černý conjecture - 1964

The length of the shortest synchronizing word for any n-state DFA is at most $(n-1)^2$.

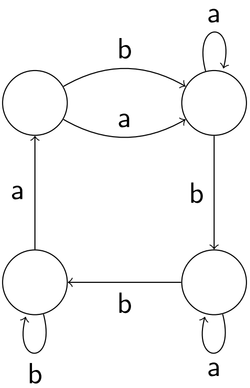
Synchronizing Words

- Given an automaton, can you find a word that brings the automaton to the same single state no-matter where it started?
- Seen practical applications in
  - biocomputing
  - model-based testing, and
  - robotics.
- The Černý conjecture is one of the longest standing conjectures in automata theory

Černý conjecture - 1964

The length of the shortest synchronizing word for any n-state DFA is at most $(n-1)^2$.

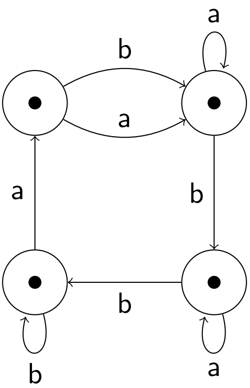
Synchronizing Words



Dose a synchronizing word exists?

Yes - bbb or aababaa

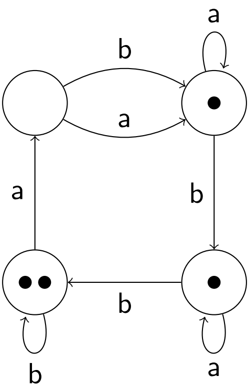
Synchronizing Words



Dose a synchronizing word exists?

Yes - bbb or **aababaa**

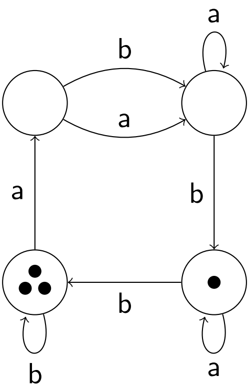
Synchronizing Words



Dose a synchronizing word exists?

Yes - bbb or **aababaa**

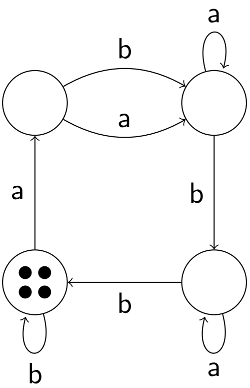
Synchronizing Words



Dose a synchronizing word exists?

Yes - bbb or **aababaa**

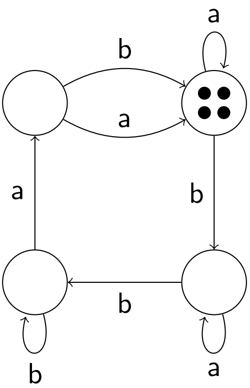
Synchronizing Words



Dose a synchronizing word exists?

Yes - bbb or **aababaa**

Synchronizing Words



Dose a synchronizing word exists?

Yes - **bbb** or **aababaa**

The three Problems

Problem 1 (Synchronization)

Given an finite automata $\mathcal{A}$, is there a synchronizing word w for $\mathcal{A}$?

Problem 2 (Short-Synchronization)

Given an finite automata $\mathcal{A}$ and a bound $k \in \mathbb{N}$, is there a synchronizing word w for $\mathcal{A}$ such that $|w| \leq k$?

Problem 3 (Subset-to-Subset Synchronization)

Given an finite automata $\mathcal{A}$ and subsets $S_{from}, S_{to} \subseteq S$, is there a word w that synchronize all states from S_{from} to a state in S_{to}?

The three Problems

Problem 1 (Synchronization)

Given an finite automata $\mathcal{A}$, is there a synchronizing word w for $\mathcal{A}$?

Problem 2 (Short-Synchronization)

Given an finite automata $\mathcal{A}$ and a bound $k \in \mathbb{N}$, is there a synchronizing word w for $\mathcal{A}$ such that $|w| \leq k$?

Problem 3 (Subset-to-Subset Synchronization)

Given an finite automata $\mathcal{A}$ and subsets $S_{from}, S_{to} \subseteq S$, is there a word w that synchronize all states from S_{from} to a state in S_{to}?

The three Problems

Problem 1 (Synchronization)

Given an finite automata $\mathcal{A}$, is there a synchronizing word w for $\mathcal{A}$?

Problem 2 (Short-Synchronization)

Given an finite automata $\mathcal{A}$ and a bound $k \in \mathbb{N}$, is there a synchronizing word w for $\mathcal{A}$ such that $|w| \leq k$?

Problem 3 (Subset-to-Subset Synchronization)

Given an finite automata $\mathcal{A}$ and subsets $S_{from}, S_{to} \subseteq S$, is there a word w that synchronize all states from S_{from} to a state in S_{to}?

Complexity

	Synchronization
DFA	NL-complete [Černý (1964); Volkov (2008)]
PFA	PSPACE-complete [Martyugin (2013)]
NFA	PSPACE-complete [Rystsov (1992); Martyugin (2013)]
	Short-Synchronization
DFA	NP-complete [Eppstein (1990)]
PFA	PSPACE-complete [Martyugin (2013)]
NFA	PSPACE-complete [Martyugin (2013)]
	Subset-to-subset Synchronization
DFA	PSPACE-complete [Rystsov (1983)]
PFA	PSPACE-complete [Rystsov (1983)]
NFA	PSPACE-complete [Rystsov (1983)]

Motivation

Why partial observability

- Full knowledge about the current state is utopia
- However, no information is to pessimistic
- Synchronization where studied with a blind controller
- It allows us to synchronize more interesting systems

Our goal

- We want to give the controller some partial information about the current state
- Go from synchronizing word to synchronizing strategy
- Determine complexity bounds in the different scenarios

Motivation

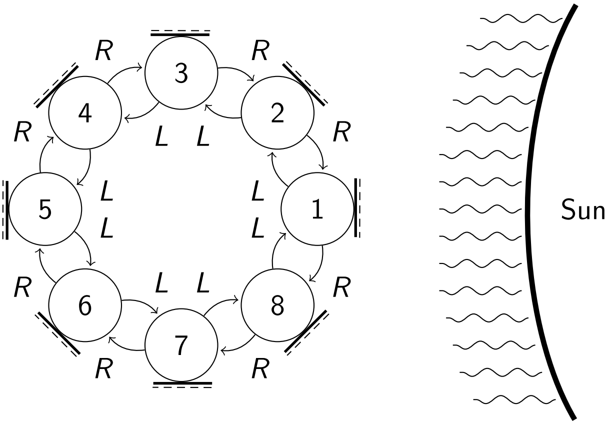
Why partial observability

- Full knowledge about the current state is utopia
- However, no information is to pessimistic
- Synchronization where studied with a blind controller
- It allows us to synchronize more interesting systems

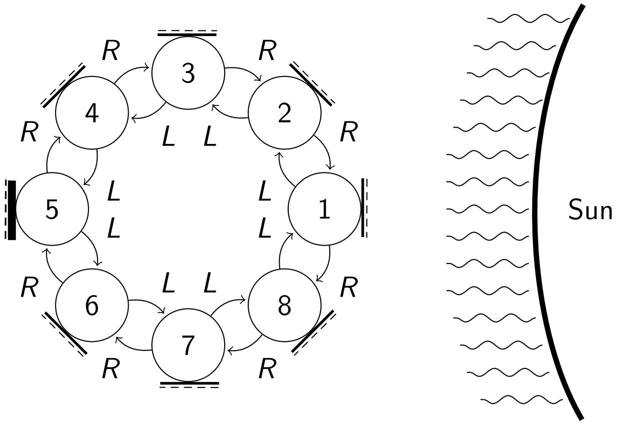
Our goal

- We want to give the controller some partial information about the current state
- Go from synchronizing word to synchronizing strategy
- Determine complexity bounds in the different scenarios

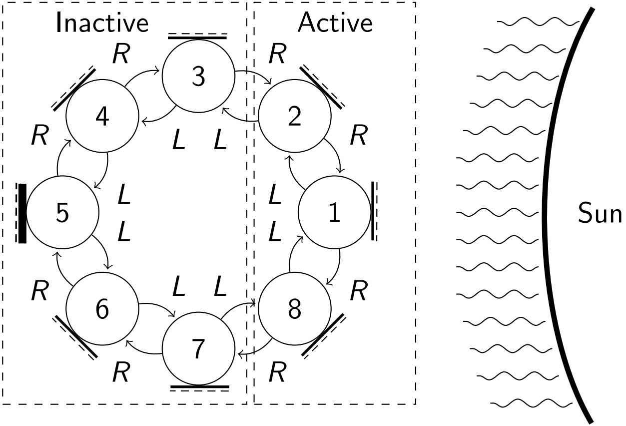
Satellite with Partial Observability



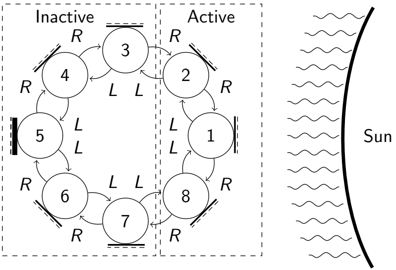
Satellite with Partial Observability



Satellite with Partial Observability



Satellite and its Synchronizing Strategy



Synchronizing strategy that brings the satellite to state 1:

if Inactive **then**
  repeat rotate L
  until Active;
else repeat rotate R
   until Inactive;
   rotate L;
endif
rotate L;

Labeled Transition System

Definition (LTS with Partial Observability)

A quintuple $T = (S, Act, \rightarrow, \mathcal{O}, \gamma)$ where

- S is a set of states,
- Act is an action alphabet,
- $\rightarrow \subseteq S \times Act \times S$ is the transition relation,
- $\mathcal{O}$ is a nonempty set of observations, and
- $\gamma : S \rightarrow \mathcal{O}$ is a mapping from state to observation.

Path and Strategy

Definition (Path)

A *path* is a finite sequence $\pi = s_1 a_1 s_2 a_2 \ldots a_{n-1} s_n$, $last(\pi) = s_n$ and the set of all finite paths in T is denoted by $paths(T)$.

Definition (Observation sequence)

The observation sequence of π is $\gamma(\pi) = \gamma(s_1)\gamma(s_2)\ldots\gamma(s_n)$.

Definition (Strategy)

A *strategy* on T is a function $\delta : \mathcal{O}^+ \to Act \cup \{done\}$ where $done \notin Act$. We only consider terminating strategies.

Path and Strategy

Definition (Path)

A *path* is a finite sequence $\pi = s_1 a_1 s_2 a_2 \ldots a_{n-1} s_n$, $last(\pi) = s_n$ and the set of all finite paths in T is denoted by $paths(T)$.

Definition (Observation sequence)

The observation sequence of π is $\gamma(\pi) = \gamma(s_1)\gamma(s_2)\ldots\gamma(s_n)$.

Definition (Strategy)

A *strategy* on T is a function $\delta : \mathcal{O}^+ \to Act \cup \{done\}$ where $done \notin Act$. We only consider terminating strategies.

Synchronizing Strategy

Definition (Strategy)

A *strategy* on T is a function $\delta : \mathcal{O}^+ \to \mathit{Act} \cup \{\mathit{done}\}$ where $\mathit{done} \notin \mathit{Act}$. We only consider terminating strategies.

Definition (Following a strategy)

The set of all paths starting from $X \subseteq S$ that follows δ

$$\begin{aligned}\delta[X] = \{\pi = s_1 a_1 \dots a_{n-1} s_n \in \mathit{paths}(T) \mid \\ s_1 \in X, \delta(\gamma(\pi)) = \mathit{done} \text{ and} \\ \delta(\gamma(s_1 a_1 s_2 a_2 \dots s_i)) = a_i \text{ for all } i, 1 \leq i < n \} .\end{aligned}$$

Definition (Synchronizing)

A strategy δ is **synchronizing** if $\mathit{last}(\delta[S])$ is a singleton set.

Knowledge Game

Problem: Synchronizing strategy $\delta : \mathcal{O}^+ \rightarrow \mathit{Act} \cup \{\mathit{done}\}$

- Two-player game played on a graph where each node represents a set of states called a belief
- *Player 1* plays by proposing an action
- *Player 2* then determines which of the possible next beliefs the play continues from
- *Player 1* wins the knowledge game if she has a **memory-less** strategy so that any play under this strategy reaches the same singleton belief $\{s\}$

Knowledge Game

Problem: Synchronizing strategy $\delta : \mathcal{O}^{+} \rightarrow \mathit{Act} \cup \{\mathit{done}\}$

- Two-player game played on a graph where each node represents a set of states called a belief
- *Player 1* plays by proposing an action
- *Player 2* then determines which of the possible next beliefs the play continues from
- *Player 1* wins the knowledge game if she has a **memory-less** strategy so that any play under this strategy reaches the same singleton belief $\{s\}$

Knowledge Game

Definition (Split function)

The function splits set of states $X \subseteq S$ into observations

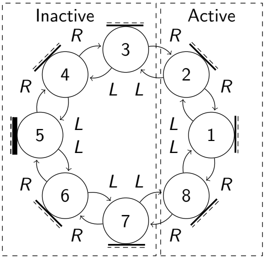
$$split(X) = \{\{s \in X \mid \gamma(s) = o\} \mid o \in \mathcal{O}\} \setminus \emptyset$$

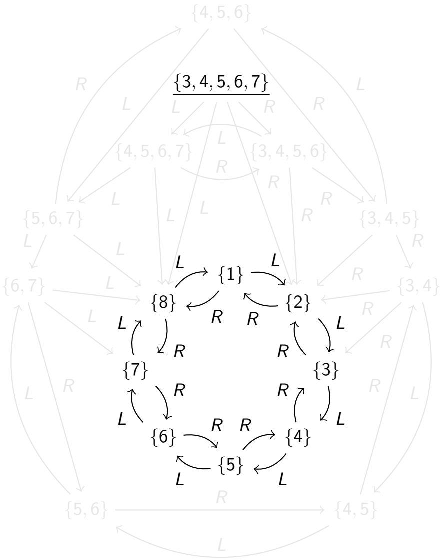
Definition (Knowledge Game)

Given an LTSP $T = (S, Act, \rightarrow, \mathcal{O}, \gamma)$, the corresponding knowledge game is a quadruple $G(T) = (\mathcal{V}, \mathcal{I}, Act, \Rightarrow)$ where

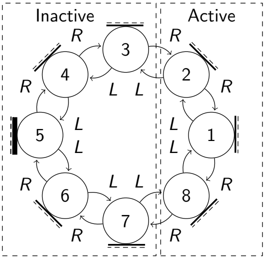
- $\mathcal{V} = \{V \in 2^S \setminus \emptyset \mid V = split(V)\}$ is the set of all beliefs that cannot be split,
- $\mathcal{I} = split(S)$ is the set of initial beliefs, and
- $\Rightarrow \subseteq \mathcal{V} \times Act \times \mathcal{V}$ is the transition relation, such that $V_1 \overset{a}{\Rightarrow} V_2$ iff $V_2 \in split(succ(V_1, a))$.

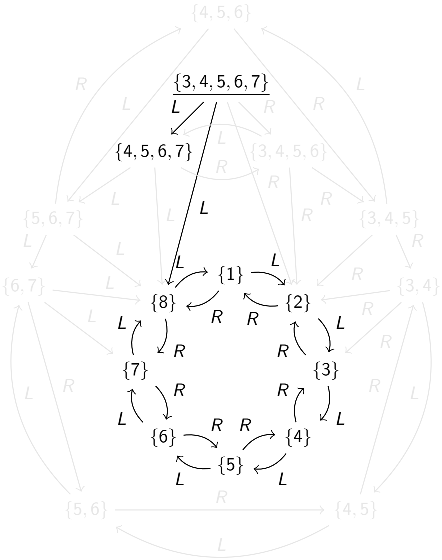
Knowledge Game - Example



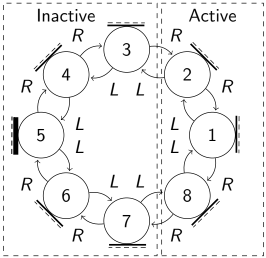


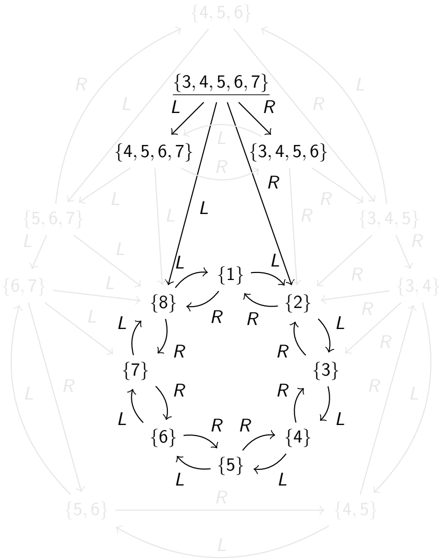
Knowledge Game - Example



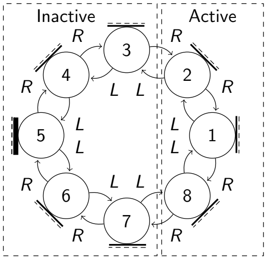


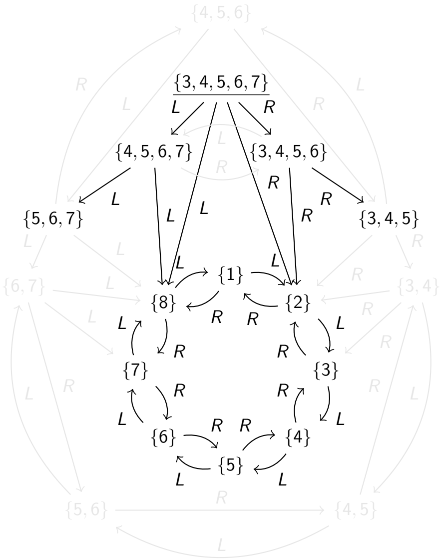
Knowledge Game - Example



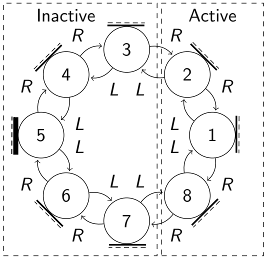


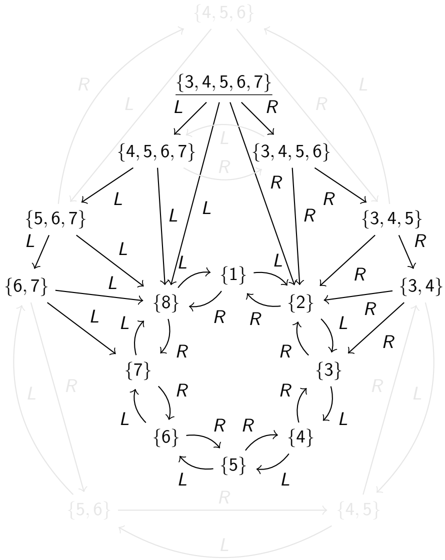
Knowledge Game - Example



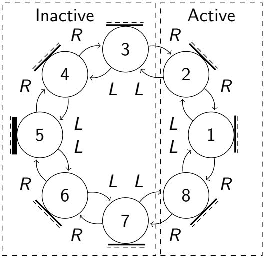


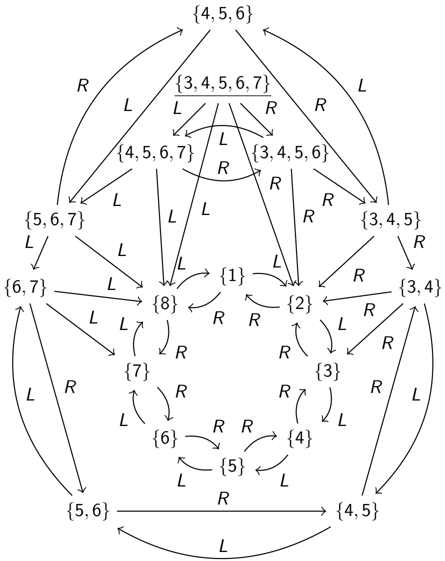
Knowledge Game - Example





Knowledge Game - Example





Knowledge Game

Theorem (Correctness)

Then Player 1 wins the knowledge game $G(T)$ iff there is a synchronizing strategy for T.

Theorem (EXPTIME - Upper-bound)

The synchronization, short-synchronization and subset-to-subset synchronization problems for NFA are in EXPTIME.

The proof is done by exploring in polynomial time the underlying exponentially large graph of the knowledge game.

Aggregated Knowledge Graph

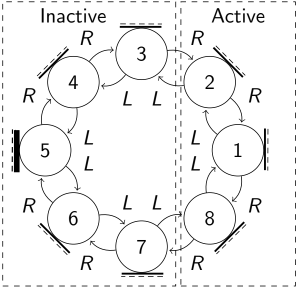
- Every note is a **set of beliefs** called a configuration
- We ask a reachability question from the configuration of the initial beliefs to a configuration with only one singleton belief.

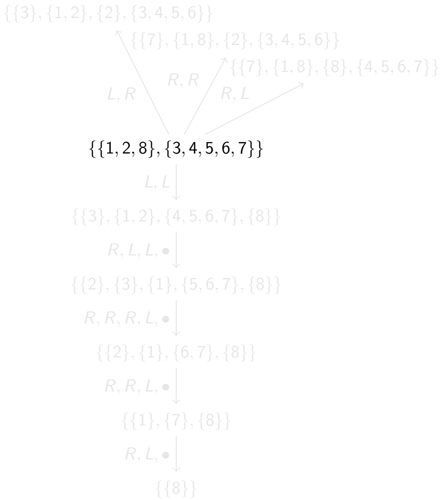
Definition (Aggregated Knowledge Graph)

Let $G(T) = (\mathcal{V}, \mathcal{I}, Act, \Rightarrow)$ be a knowledge game. The aggregated knowledge graph is a tuple $AG(G(T)) = (\mathscr{C}, \mathcal{C}_0, \rightrightarrows)$ where

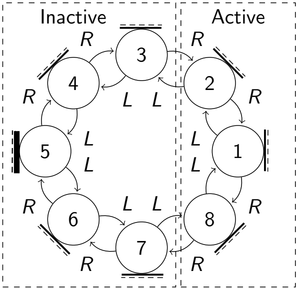
- $\mathscr{C} = 2^{\mathcal{V}} \setminus \emptyset$ is the set of configurations,
- $\mathcal{C}_0 = \mathcal{I}$ is the initial configuration, and
- $\rightrightarrows \subseteq \mathscr{C} \times \mathscr{C}$ is the transition relation such that $\mathcal{C}_1 \rightrightarrows \mathcal{C}_2$, is possible if for every $V \in \mathcal{C}_1$ there is an action $a_V \in Act \cup \{\bullet\}$ such that $V \overset{a_V}{\Longrightarrow} V'$ for at least one V' ending in $\mathcal{C}_2 = \{V' \mid V \in \mathcal{C}_1 \text{ and } V \overset{a_V}{\Longrightarrow} V'\}$.

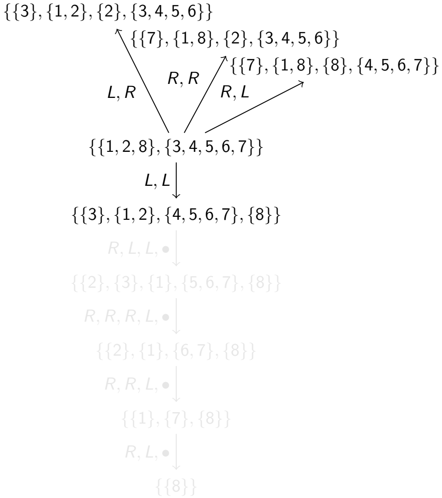
Aggregated Knowledge Graph - Example



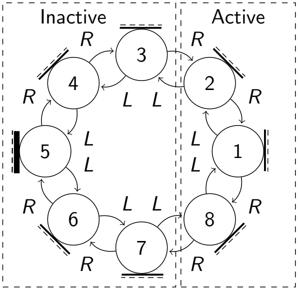


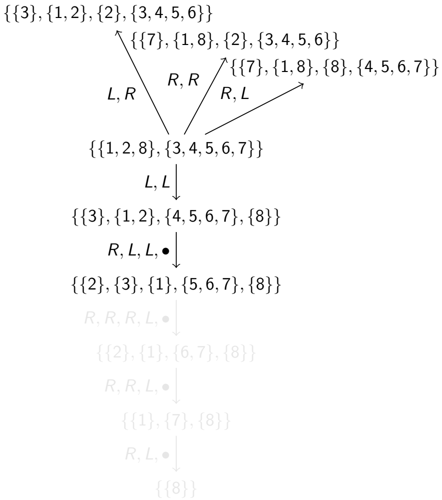
Aggregated Knowledge Graph - Example



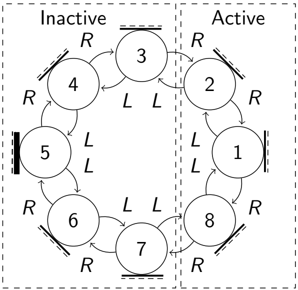


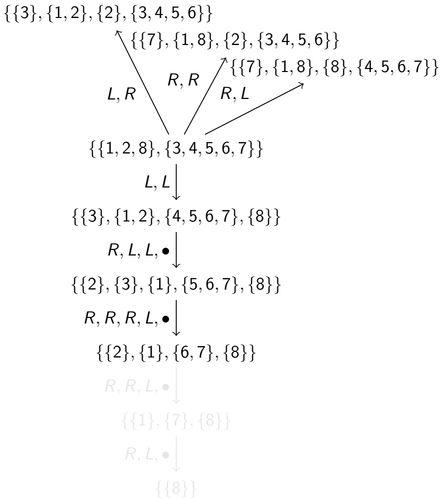
Aggregated Knowledge Graph - Example



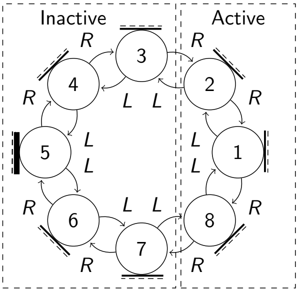


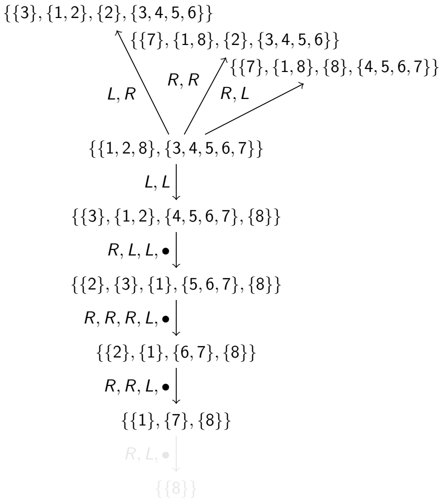
Aggregated Knowledge Graph - Example



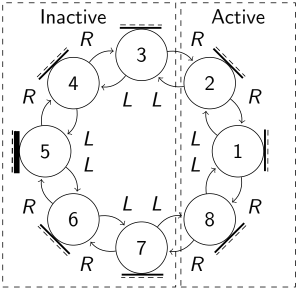


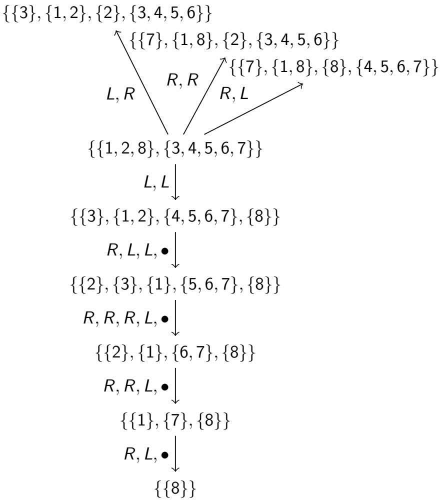
Aggregated Knowledge Graph - Example





Aggregated Knowledge Graph - Example





Aggregated Knowledge Graph

Theorem (Correctness)

Let $AG(G(T))$ be the aggregated knowledge graph over the knowledge game $G(T)$. Then is a path from the configuration of initial beliefs $\mathcal{C}_0$ to some configuration with only one singleton belief $\{\{s\}\}$ if and only if Player 1 wins the knowledge game $G(T)$.

Theorem (PSPACE Upper-bound)

The synchronization, short-synchronization and subset-to-subset synchronization problems for DFA and PFA are decidable in PSPACE.

The proof follows from the fact that the next configuration in path can be guessed in polynomial space as the size of configurations do not grow exponentially.

Aggregated Knowledge Graph

Theorem (Correctness)

Let $AG(G(T))$ be the aggregated knowledge graph over the knowledge game $G(T)$. Then is a path from the configuration of initial beliefs $\mathcal{C}_0$ to some configuration with only one singleton belief $\{\{s\}\}$ if and only if Player 1 wins the knowledge game $G(T)$.

Theorem (PSPACE Upper-bound)

The synchronization, short-synchronization and subset-to-subset synchronization problems for DFA and PFA are decidable in PSPACE.

The proof follows from the fact that the next configuration in path can be guessed in polynomial space as the size of configurations do not grow exponentially.

Aggregated Knowledge Graph

Theorem (NL upper-bound)

The synchronization problem on DFA is in NL.

In the proof we use a generalization an old result saying that all pairs of states in the system can synchronize if an only if all states can synchronize.

Theorem (NP upper-bound)

The short-synchronization problem on DFA is in NP.

To proove this we bound the length of the shortest synchronizing strategy in a DFA to be at most $(n-1)n^2$ where n is the number of states.

Aggregated Knowledge Graph

Theorem (NL upper-bound)

The synchronization problem on DFA is in NL.

In the proof we use a generalization an old result saying that all pairs of states in the system can synchronize if an only if all states can synchronize.

Theorem (NP upper-bound)

The short-synchronization problem on DFA is in NP.

To proove this we bound the length of the shortest synchronizing strategy in a DFA to be at most $(n-1)n^2$ where n is the number of states.

Complexity Lower-Bounds

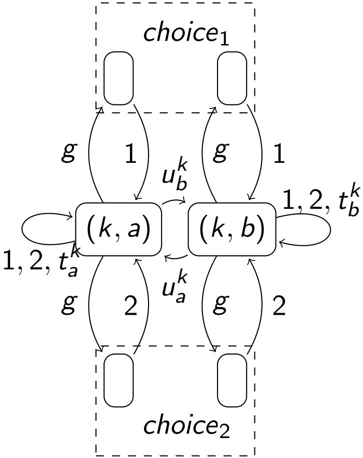
Theorem

The subset-to-subset synchronization problem is EXPTIME-hard for NFA.

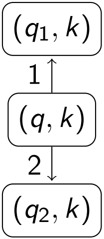
- Reduction from acceptance problem for alternating liner bounded automaton over the binary alphabet $\{a, b\}$
- We assume that existential and universal choices do not change the current head position or the tape content
- We have special deterministic states for tape manipulation
- We use the three observations $\{default, choice_1, choice_2\}$

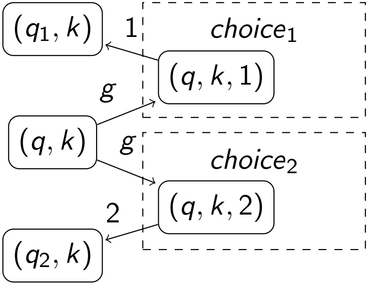
Translation

- Encoding of tape cell position k
- The t_a^k and t_b^k actions are used as tests content
- where u_a^k and u_b^k are used to update content



Translation





- Existential control state q
- The strategy can select the action 1 or 2
- All states belong to *default*

- Universal control state q
- Strategy can propose only the action g
- The choice of the nondeterminism is revealed by the observation

Translation

- From the accepting state (q_{acc}, k) and the any cell state (k, a) and (k, b) we allow it to enter a new state *sink*
- Synchronization only possible in *sink*
- The S_{from} states $(q_0, 1)$ and tape cells corresponding to initial content
- This forces all runs under the synchronizing strategy to correspond to an accepting run of the ALBA

Theorem

The synchronization and short-synchronization problems are EXPTIME-hard for NFA.

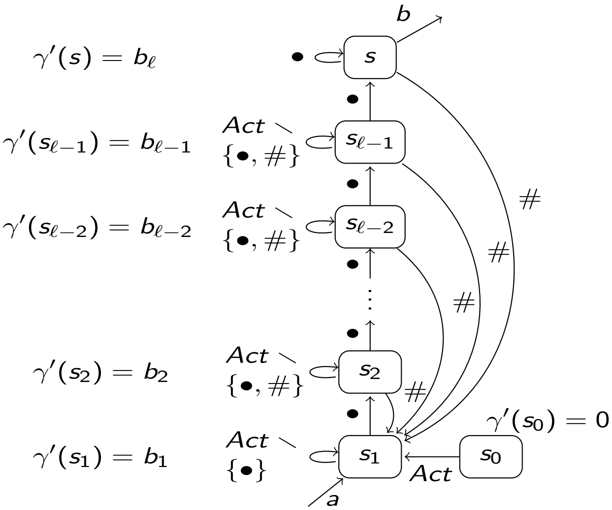
Two Observations are Enough

Theorem (Two observations)

The synchronization, short-synchronization and subset-to-subset synchronization problems on DFA, PFA and NFA are polynomial-time reducible to the equivalent problems with only two observations.

- General reduction from arbitrary number of observations to two observations
- Increasing the size of the system by only a logarithmic factor
- Encode an observation into a sequence of observations over a binary alphabet

Two Observations - Construction



New states for every $s \in S$ where $\gamma(s) = b_1 b_2 \dots b_\ell$

Complexity overview

		Classical synchronization One observation	Partial Observability No restriction on observations
Synch. Prob.	DFA	NL-complete	**NL-complete**
	PFA	PSPACE-complete	**PSPACE-complete**
	NFA	PSPACE-complete	**EXPTIME-complete**
Short-synch.	DFA	NP-complete	**NP-complete**
	PFA	PSPACE-complete	**PSPACE-complete**
	NFA	PSPACE-complete	**EXPTIME-complete**
Subset-to-sub.	DFA	PSPACE-complete	**PSPACE-complete**
	PFA	PSPACE-complete	**PSPACE-complete**
	NFA	PSPACE-complete	**EXPTIME-complete**

Summary and Future Work

Summary

- We defined synchronization under partial observability
- We found matching lower and upper-bounds for all problems
- In case of nondeterministic systems the problems moved to a higher complexity class
- We show that the results hold for systems with just two observations

Future work

- Quantitative synchronization - “cheapest” synchronizing strategy
- Restriction on the sequences of actions or order of states visited

Ananichev, D., & Volkov, M. (2004). Synchronizing monotonic automata. *Theoretical Computer Science*, *327*(3), 225 - 239.

Ashby, W. R. (n.d.). *An introduction to cybernetics.* New York,J. Wiley,. Retrieved from http://www.biodiversitylibrary.org/item/26977 (http://www.biodiversitylibrary.org/bibliography/5851)

Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z., & Shapiro, E. (2003, March). Dna molecule provides a computing machine with both data and fuel. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(5), 2191–2196.

Boel, T. (2013, 8. March). *Årsag til fejl på aalborg-satellit: Solcellerne vendte væk fra solen.* Ingeniøren (Weekly national news magazine about engineering). Retrieved from http://http://ing.dk/artikel/aarsag-til-fejl-paa-aalborg-satellit-solcellerne-vendte-vaek-fra-solen-156828

Cassez, F., David, A., Larsen, K. G., Lime, D., & Raskin, J.-F. (2007). Timed control with observation based and stuttering invariant strategies. In *Automated technology for verification and analysis* (Vol. 4762, p. 192-206). Springer.

Černý, J. (1964). Poznámka k. homogénnym experimentom s