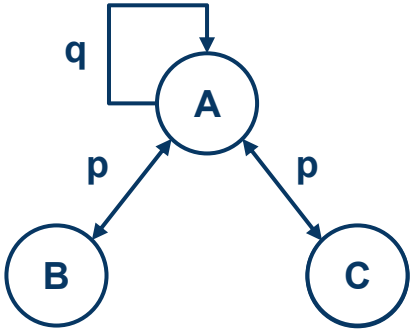
A Unified Method to Efficiently Verify Opacity of Discrete-Timed Automata

Julian Klein[1], Kuize Zhang[2], Sabine Glesner[1]
[1]Software and Embedded Systems Engineering, TU Berlin, Berlin, Germany
[2]School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China

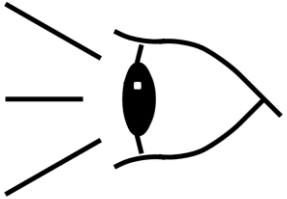
International Conference on Formal Engineering Methods
November 13, 2025

Opacity Framework



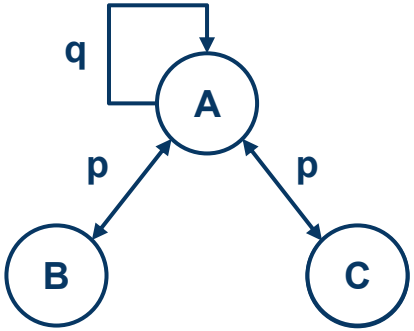
Finite Automaton

Observable Behavior



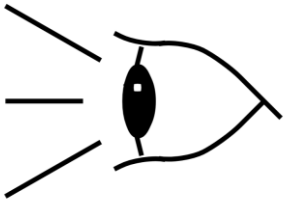
Intruder

Opacity Framework



Finite Automaton

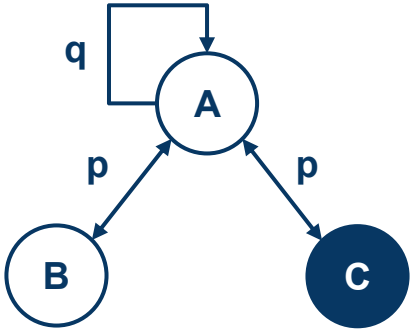
Observable Behavior



Intruder

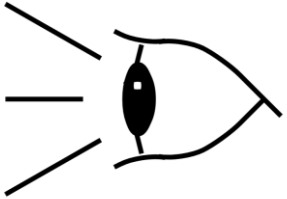
- Intruder **observes** system

Opacity Framework



Finite Automaton

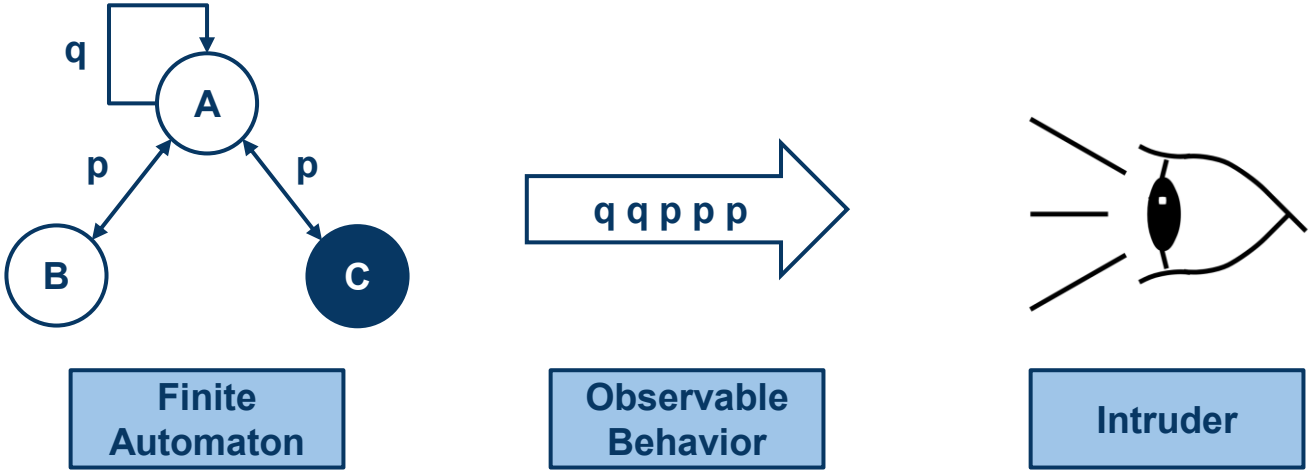
Observable Behavior



Intruder

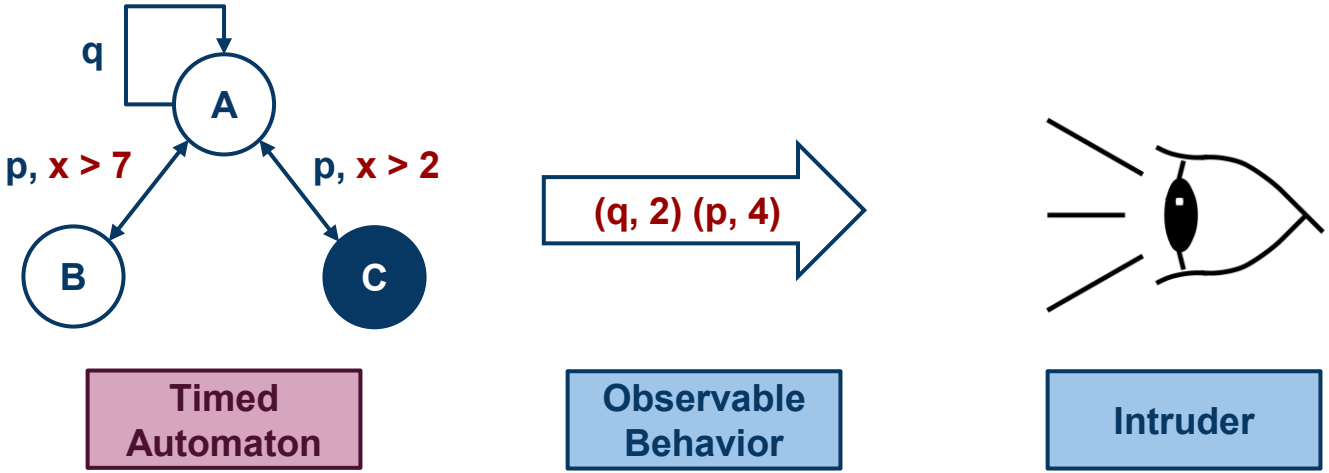
- Intruder **observes** system
- Set of **secret** states, for example {C}

Opacity Framework



- Intruder **observes** system
- Set of **secret** states, for example {C}
- **Opacity**: can intruder tell whether **secret** states **are/were active**

Timed Opacity Framework



- Intruder **observes** system
- Set of **secret** states, for example {C}
- **Opacity**: can intruder tell whether **secret** states **are/were active**

Timed Opacity

Timed Opacity Notions:

- Initial-Location Timed Opacity (**ILTO**)

➔ Intruder cannot deduce whether **initial** state was secret

- Current-Location Timed Opacity (**CLTO**)

➔ Intruder cannot deduce whether **current** state is secret

- Infinite-Step Timed Opacity (**ISTO**)

➔ Intruder cannot deduce whether a **past** state was secret

- K-Step Timed Opacity (**KSTO**)

➔ Intruder cannot deduce whether a **past** state was secret **within the past K observations**

Goal

- Verify **any** kind of **opacity notion** on **any** given **timed automaton**!

Goal

- Verify **any** kind of **opacity notion** on **any** given **timed automaton**!

Problem: in general **undecidable** (Cassez, 2009)

Literature: Verifying Timed Opacity

Literature: Verifying Timed Opacity

Restrict class of considered systems to subclasses of TA

➔ Only very specific systems can be verified

An et al. (2024), André et al. (2024), Zhang (2024), Wang et al. (2018, 2021)

Literature: Verifying Timed Opacity

Restrict class of considered systems to subclasses of TA

➔ Only very specific systems can be verified

An et al. (2024), André et al. (2024), Zhang (2024), Wang et al. (2018, 2021)

Restrict opacity notion to a weaker version

➔ Only very specific conf. guarantees can be provided

André et al. (2022, 2023, 2024)
Ammar et al. (2021)

Literature: Verifying Timed Opacity

Restrict time semantics
(Approximation using discrete-time model)
➔ Less expressive, limited scalability

An et al. (2024), André et al. (2024), Li et al. (2024), Klein et al. (2024, 2025)

Restrict class of considered systems to subclasses of TA
➔ Only very specific systems can be verified

An et al. (2024), André et al. (2024), Zhang (2024), Wang et al. (2018, 2021)

Restrict opacity notion to a weaker version
➔ Only very specific conf. guarantees can be provided

André et al. (2022, 2023, 2024)
Ammar et al. (2021)

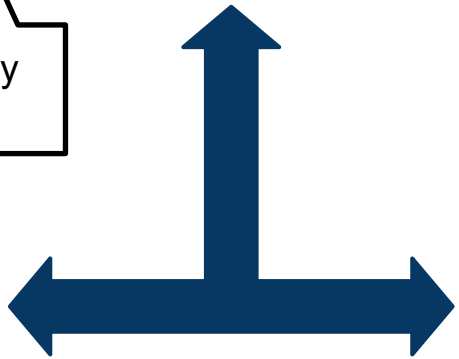
Literature: Verifying Timed Opacity

Restrict time semantics
(Approximation using discrete-time model)
➔ Less expressive, limited scalability

An et al. (2024), André et al. (2024), Li et al. (2024), **Klein et al. (2024, 2025)**

Decidability results

We are trying to develop an **efficient method**



Restrict class of considered systems to subclasses of TA
➔ Only very specific systems can be verified

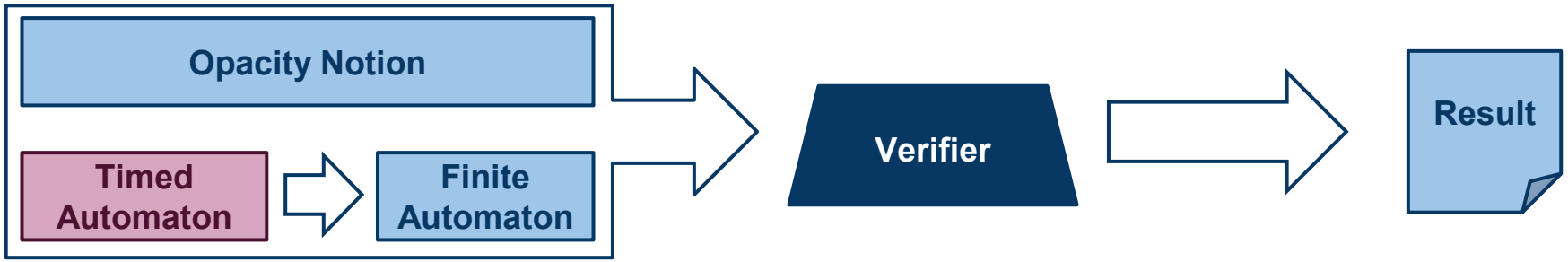
An et al. (2024), André et al. (2024), Zhang (2024), Wang et al. (2018, 2021)

Restrict opacity notion to a weaker version
➔ Only very specific conf. guarantees can be provided

André et al. (2022, 2023, 2024)
Ammar et al. (2021)

The Joy of Discrete Time

- Discrete-timed automata can be **transformed** to standard finite automata



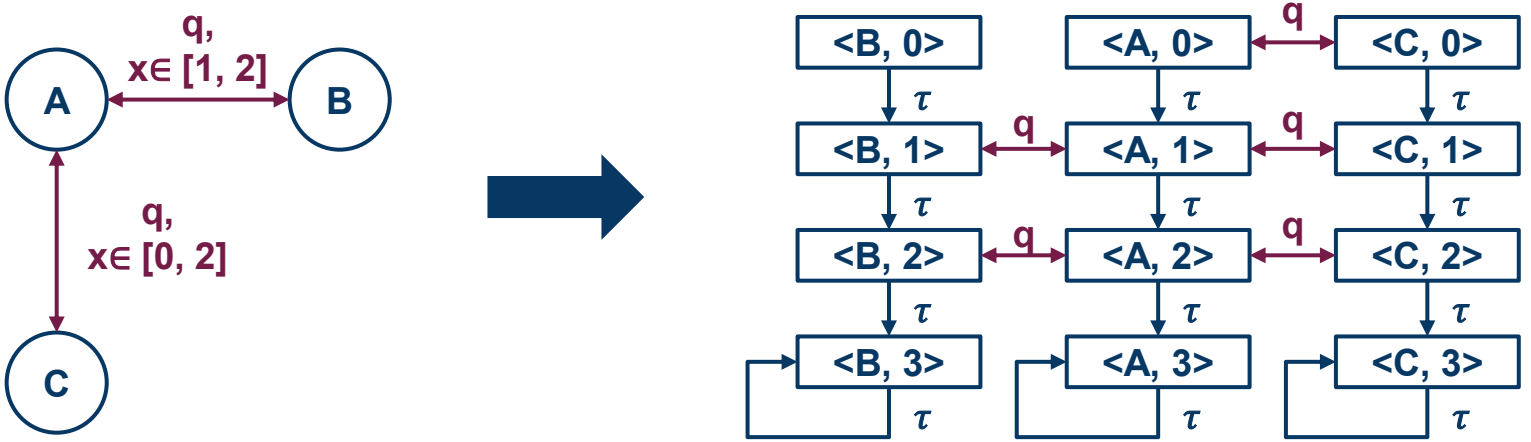
- Opacity verification of finite automata is **extensively studied**

➔ **Existing verification methods can be used**

Tick Automata

- Discrete TA can be **transformed** equivalent **tick automata** (τ-FA)
- Introduction of **tick symbol** τ to model discrete time **ticks**

Caution: Tick automata get **huge** for large input TA

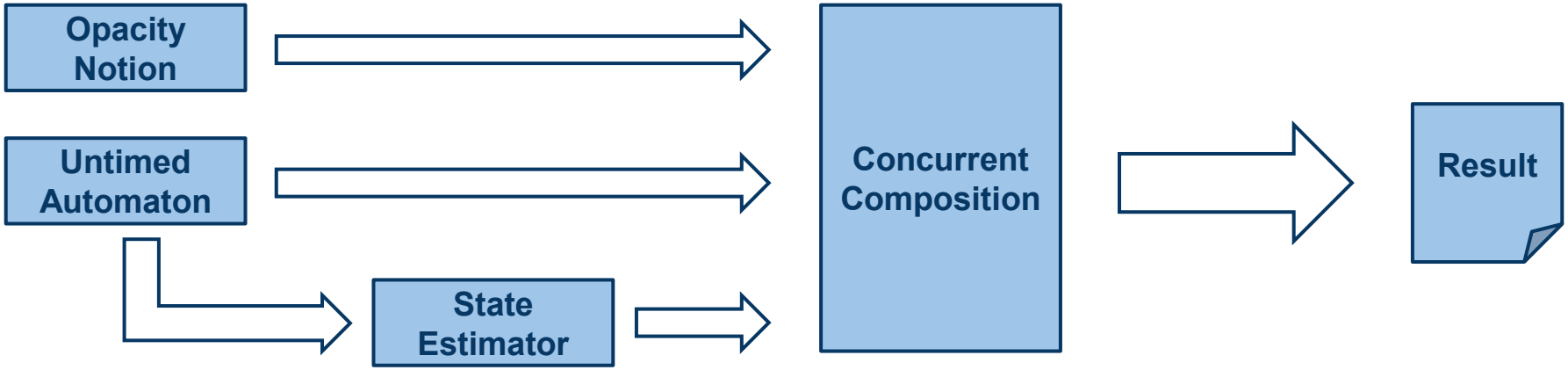


Verifying (Un)timed Opacity

Concurrent composition (CC) (Zhang 2023)

- Composition of FA and its state estimator (Observer)
- Verifying opacity notions = checking reachability properties on CC

➔ CC = Verification structure for several opacity notions!

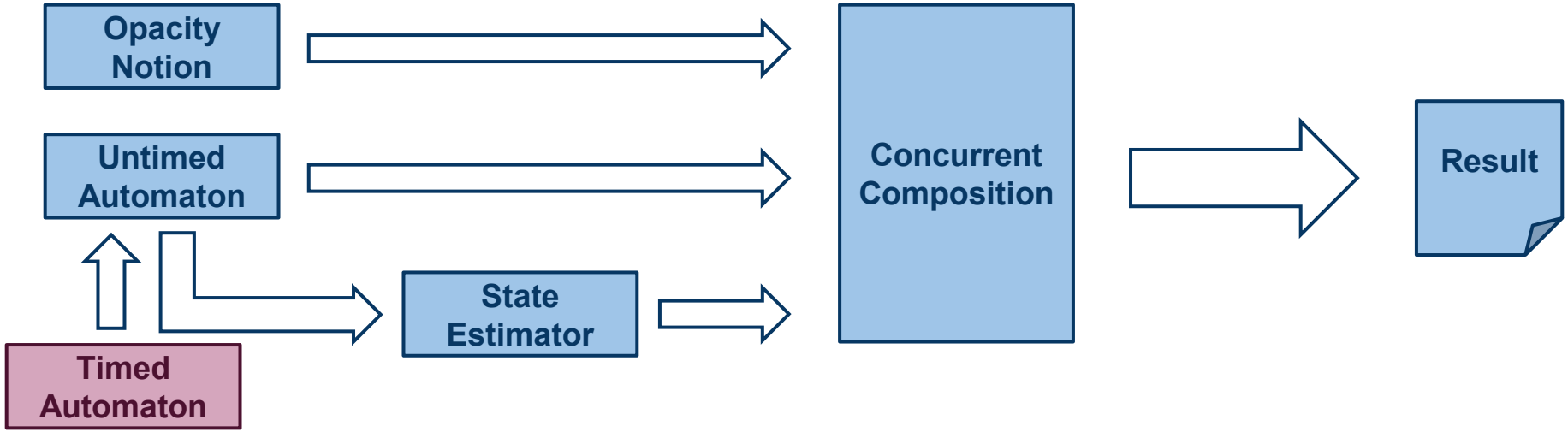


Verifying (Un)timed Opacity

Concurrent composition (CC) (Zhang 2023)

- Composition of FA and its state estimator (Observer)
- Verifying opacity notions = checking reachability properties on CC

➔ CC = Verification structure for several opacity notions!



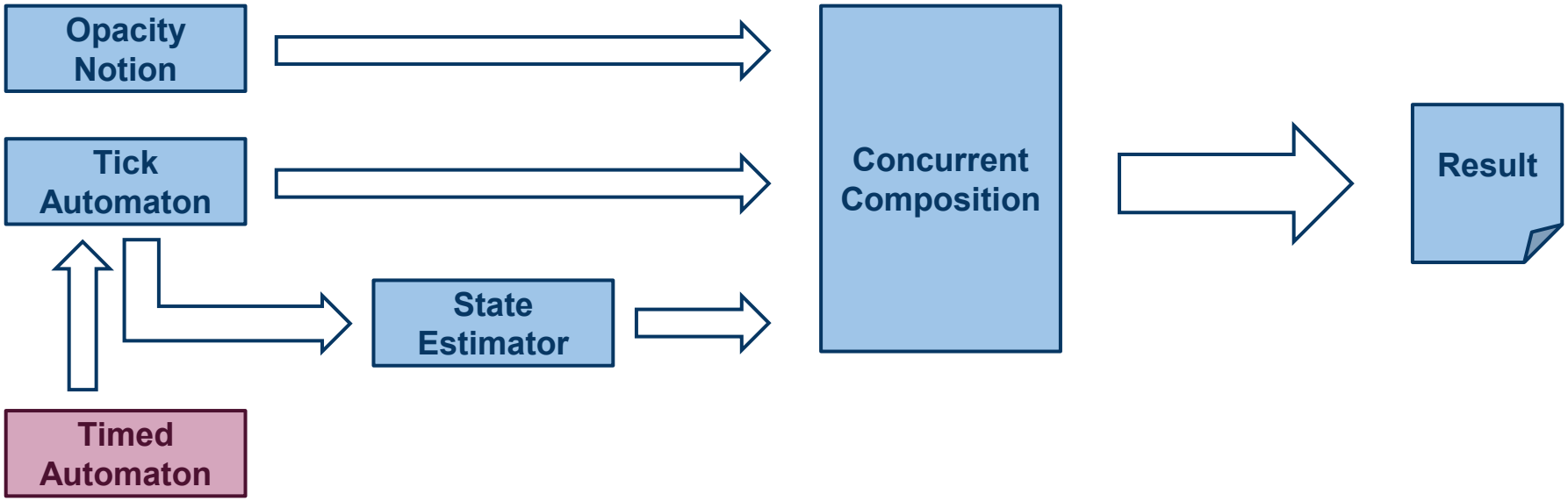
Our Method

Verifying Timed Opacity

Idea: Develop analogous timed concurrent composition

- Avoid tick automata

➔ Use purely timed formalisms

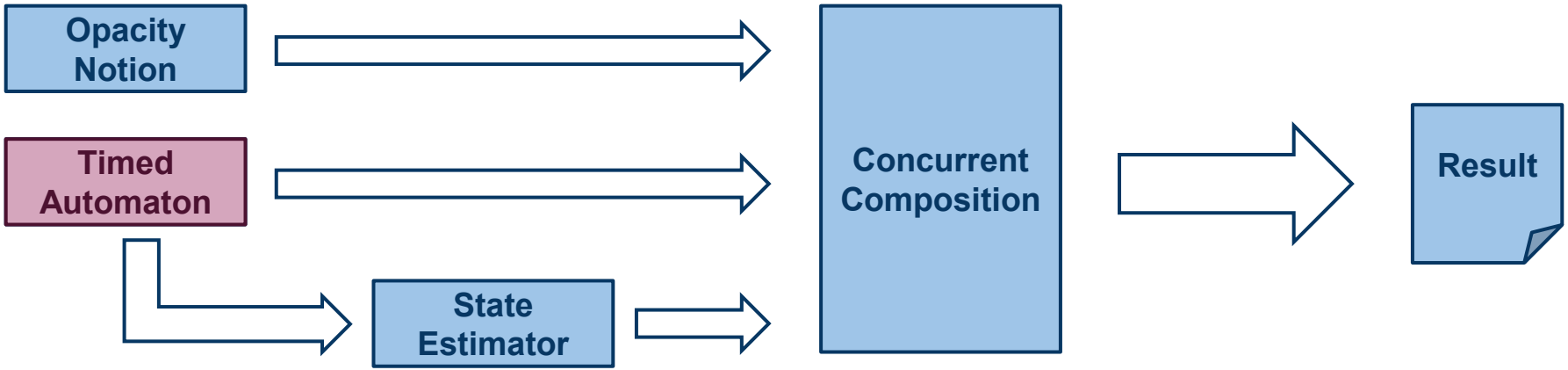


Verifying Timed Opacity

Idea: Develop analogous timed concurrent composition

- Avoid tick automata

➔ Use purely timed formalisms

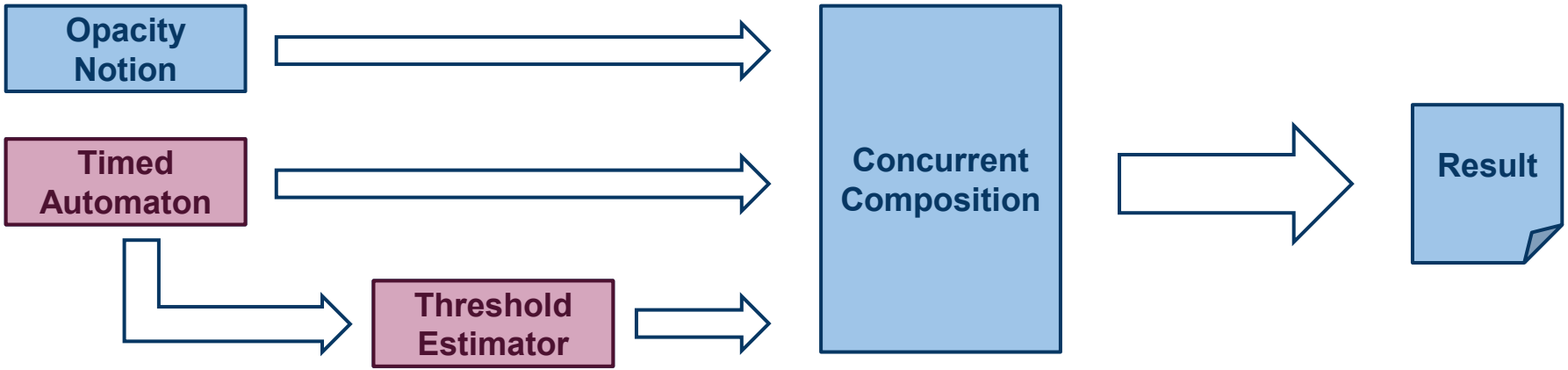


Verifying Timed Opacity

Idea: Develop analogous timed concurrent composition

- Avoid tick automata

➔ Use purely timed formalisms

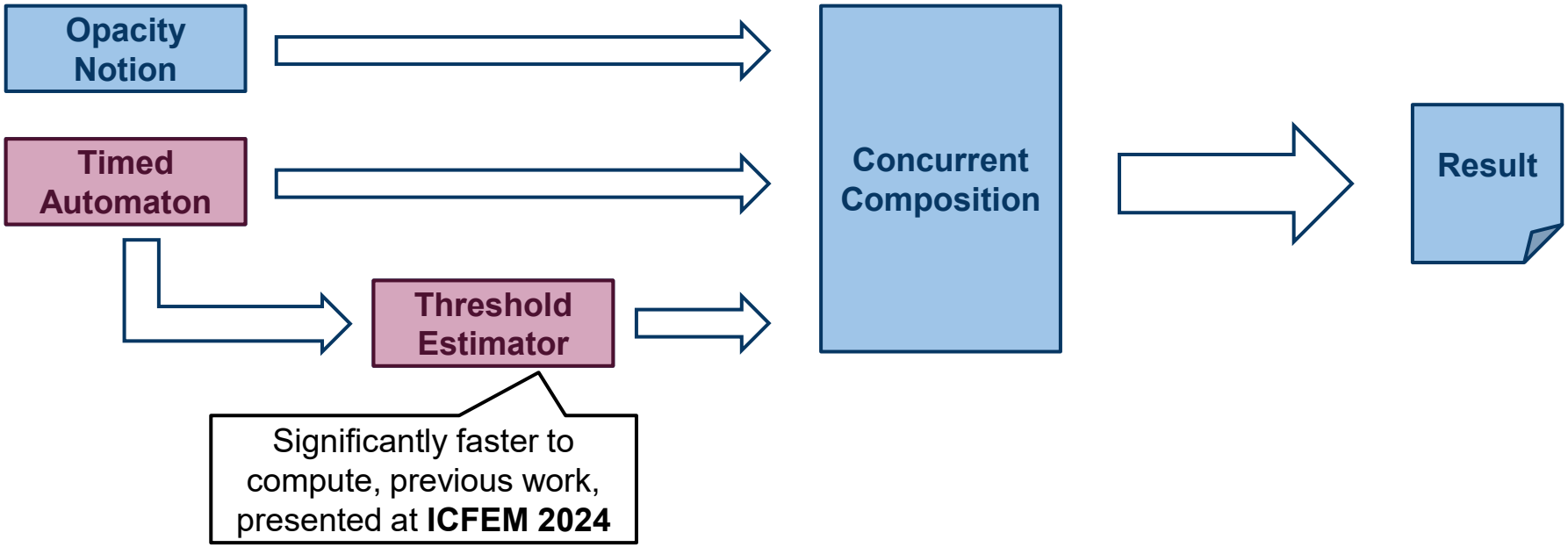


Verifying Timed Opacity

Idea: Develop analogous timed concurrent composition

- Avoid tick automata

➔ Use purely timed formalisms

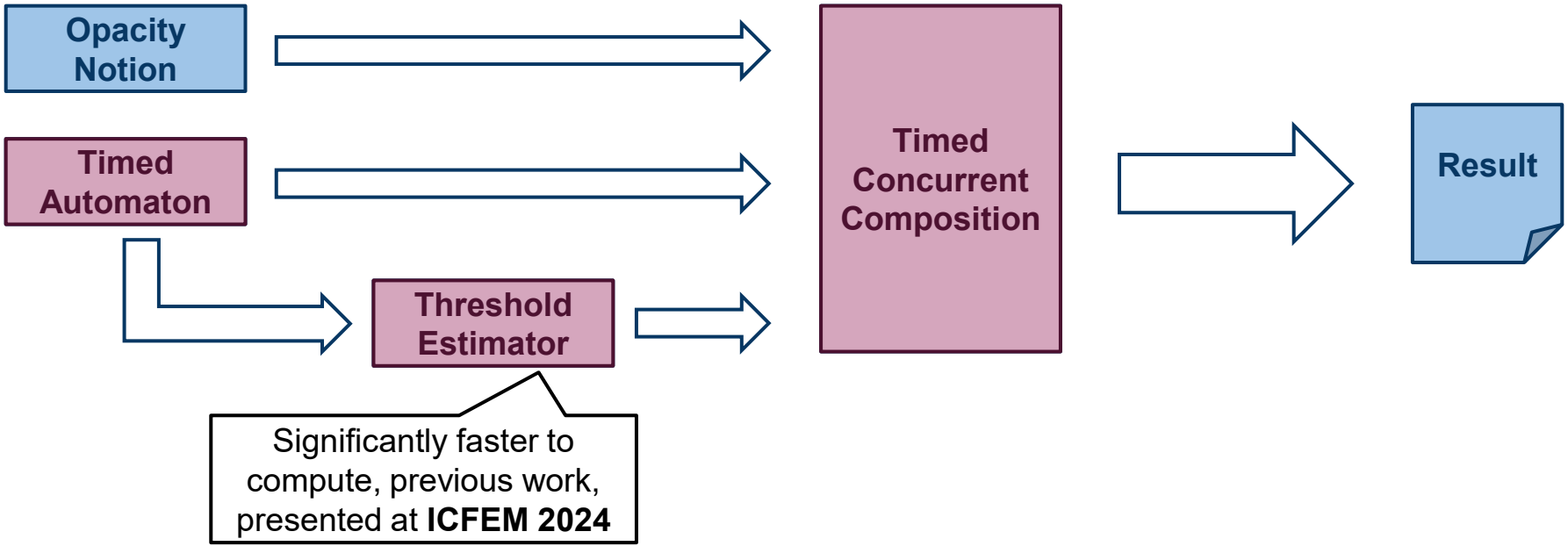


Verifying Timed Opacity

Idea: Develop analogous timed concurrent composition

- Avoid tick automata

➔ Use purely timed formalisms

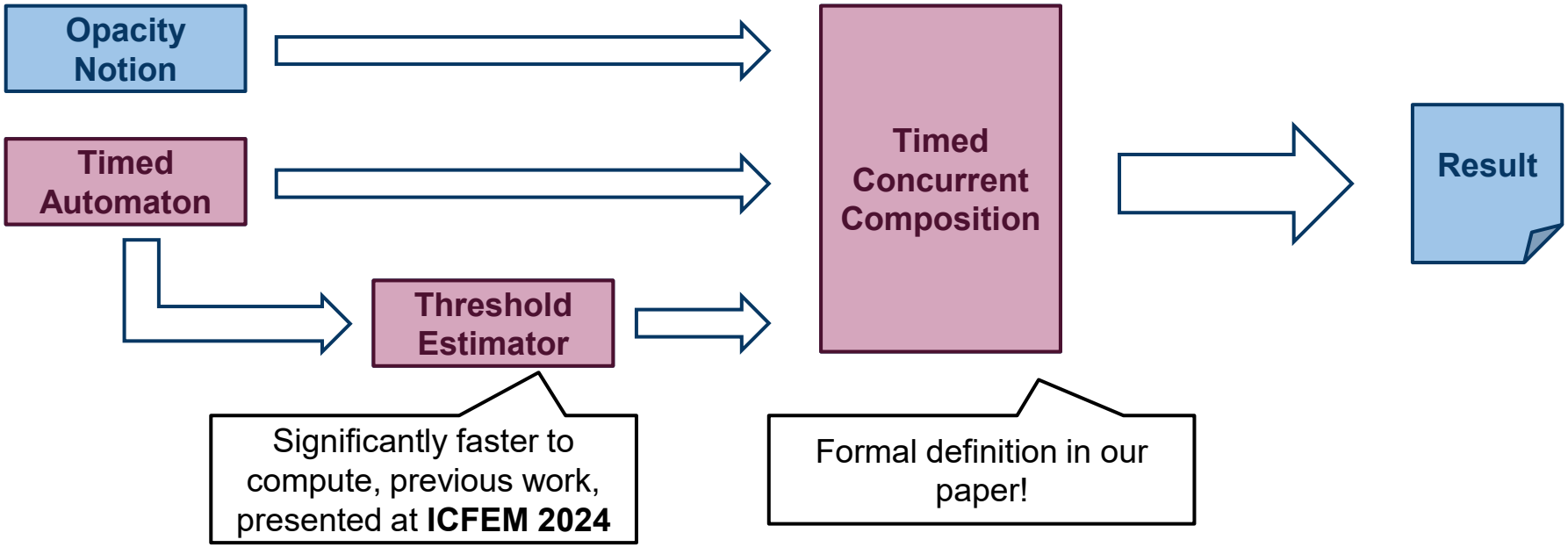


Verifying Timed Opacity

Idea: Develop analogous timed concurrent composition

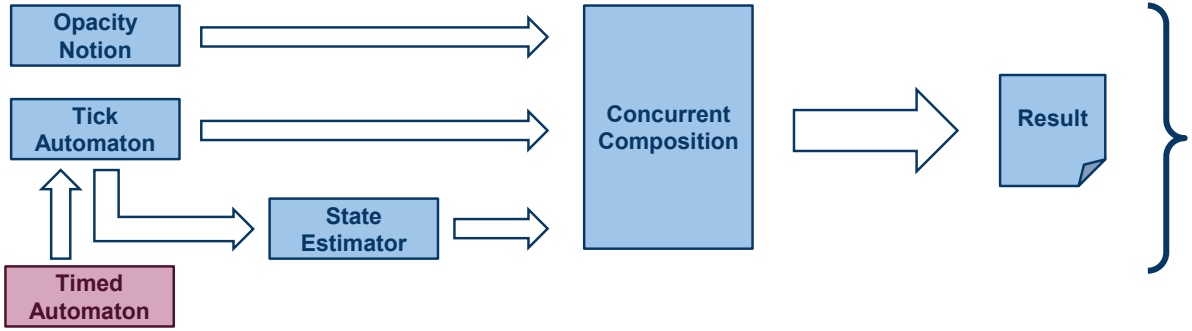
- Avoid tick automata

➔ Use purely timed formalisms



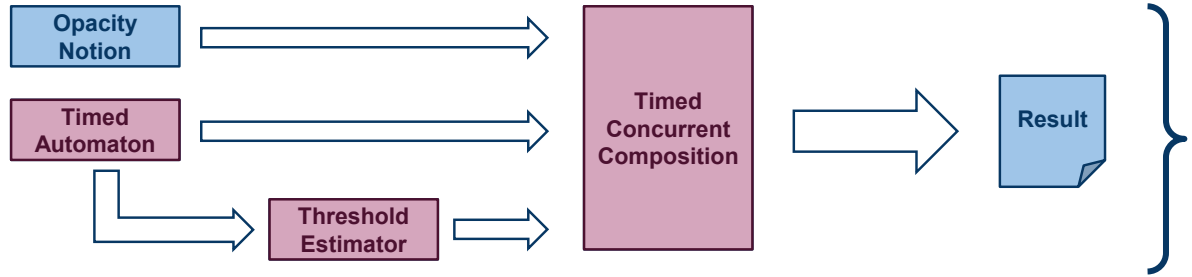
Evaluation

Implementation of Both Methods



Baseline method (CC)

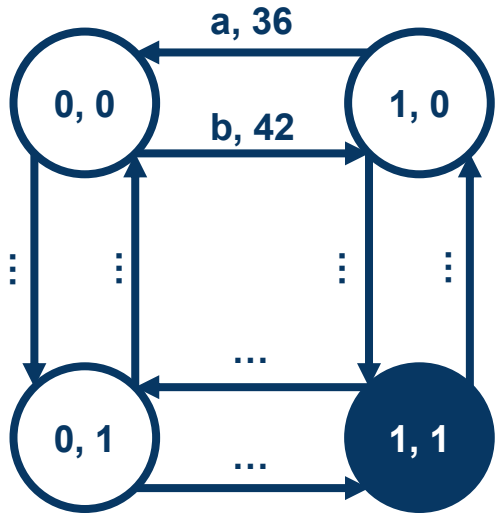
- Based on tick automata
- Uses standard concurrent composition (CC)



Our new method (TCC)

- Based on **timed** models
- Uses new, **timed** concurrent composition (TCC)

Case Study: Tracking Agents in Grid World

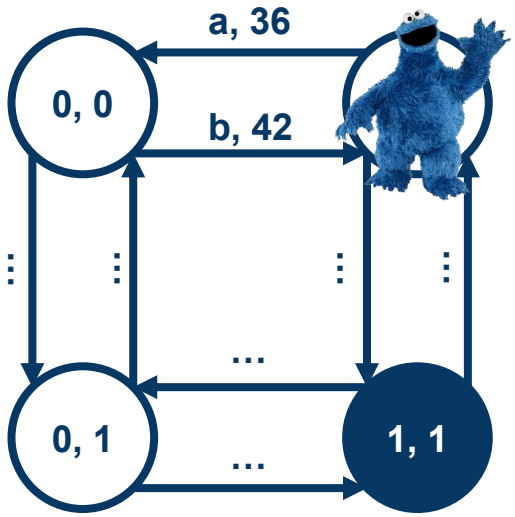


- Did agent **start** in secret location? **(ILTO)**
- Is agent **currently** in secret location? **(CLTO)**
- **Was** agent in secret location? **(ISTO)**
- **Was** agent in secret location within the last **K observations**? **(KSTO)**

Our System:

- 32 locations
- 110 transitions

Case Study: Tracking Agents in Grid World

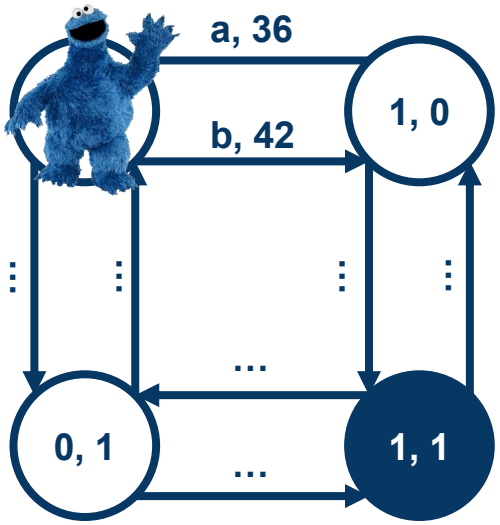


- Did agent **start** in secret location? **(ILTO)**
- Is agent **currently** in secret location? **(CLTO)**
- **Was** agent in secret location? **(ISTO)**
- **Was** agent in secret location within the last **K observations**? **(KSTO)**

Our System:

- 32 locations
- 110 transitions

Case Study: Tracking Agents in Grid World

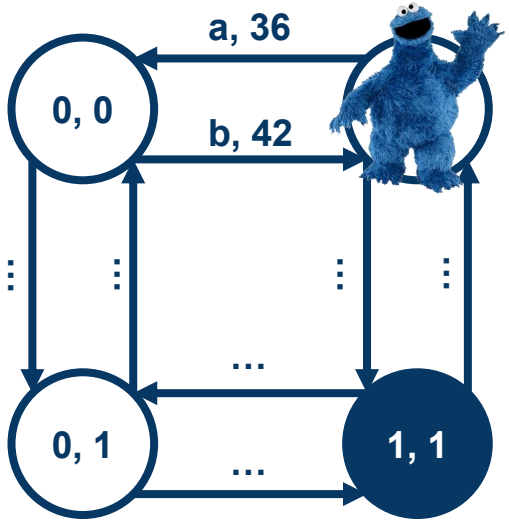


- Did agent **start** in secret location? **(ILTO)**
- Is agent **currently** in secret location? **(CLTO)**
- **Was** agent in secret location? **(ISTO)**
- **Was** agent in secret location within the last **K observations**? **(KSTO)**

Our System:

- 32 locations
- 110 transitions

Case Study: Tracking Agents in Grid World

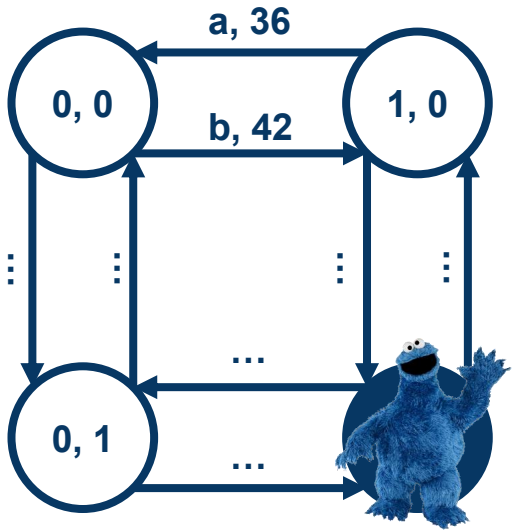


- Did agent **start** in secret location? **(ILTO)**
- Is agent **currently** in secret location? **(CLTO)**
- **Was** agent in secret location? **(ISTO)**
- **Was** agent in secret location within the last **K observations**? **(KSTO)**

Our System:

- 32 locations
- 110 transitions

Case Study: Tracking Agents in Grid World



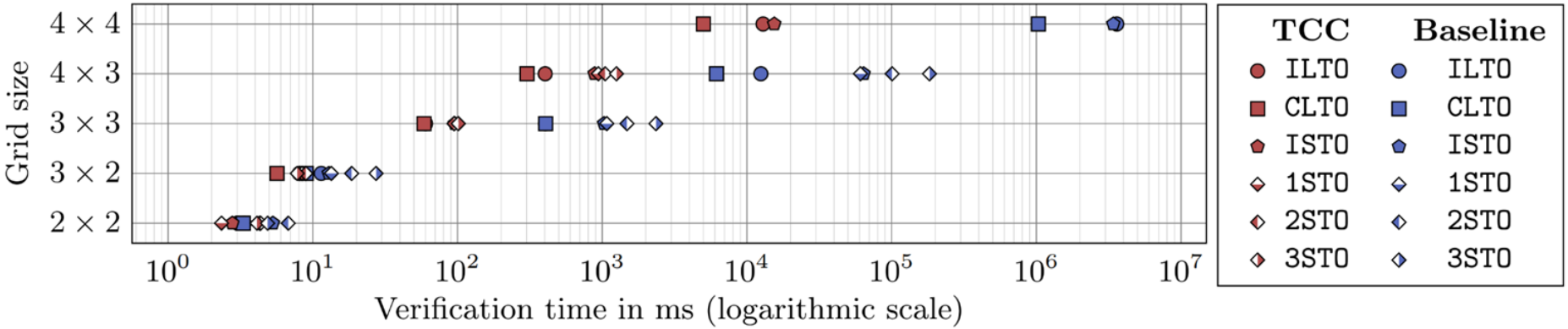
- Did agent **start** in secret location? **(ILTO)**
- Is agent **currently** in secret location? **(CLTO)**
- **Was** agent in secret location? **(ISTO)**
- **Was** agent in secret location within the last **K observations**? **(KSTO)**

Our System:

- 32 locations
- 110 transitions

Method	ILTO	CLTO	ISTO	KSTO (K > 0)
Tick automata	3min 6.66s	1min 48.23s	3min 6.25s	3min 8.67s
Our new method	35.61s	29.76s	43.20s	35.58s

Randomized Grid Worlds



- **Randomized** grids in of several sizes
- Averaged results over **100 systems of each size**

➔ **Significant reduction** in computation times across all opacity notions

➔ More significant than in case study

Conclusion

Contributions

- Definition of the **timed concurrent composition** (TCC)
- New, **unified**, and **efficient** verification for ILTO, CLTO, ISTO, and KSTO
- Implementation in fully documented open source **software prototype**
- Practical evaluation on a case study and randomized systems

Future work

- Opacity **enforcement method**, based on TCC
- Extension to settings with **multiple intruders**
- Extension to **other** opacity notions
- Testing on more **real-world applications**

Conclusion

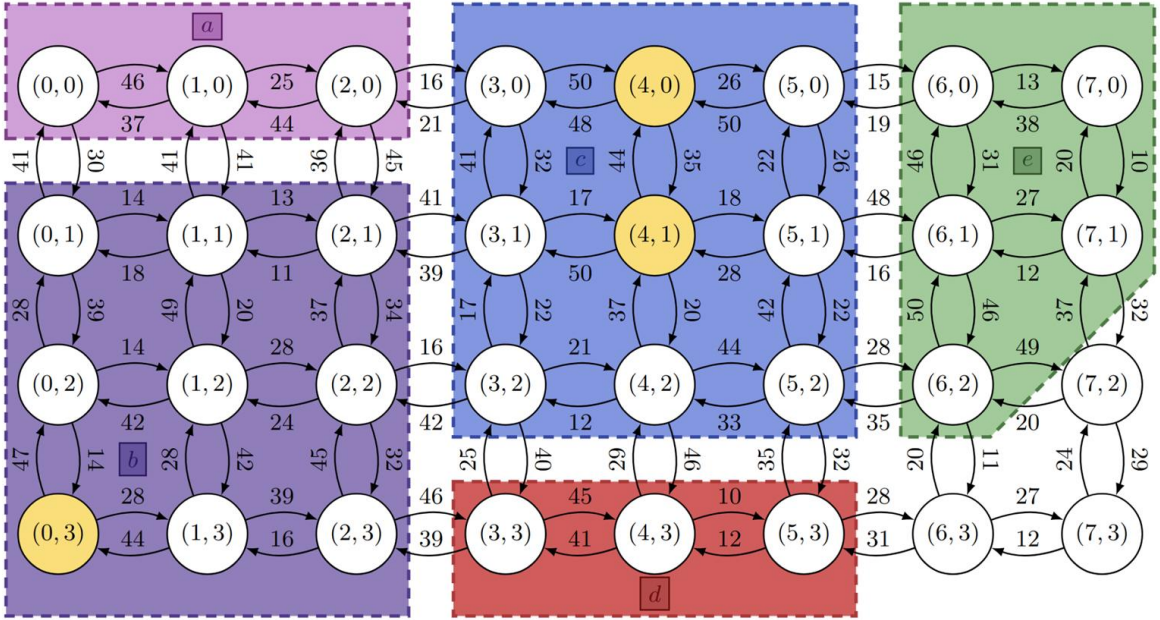
Contributions

- Definition of the **timed concurrent composition** (TCC)
- New, **unified**, and **efficient** verification for ILTO, CLTO, ISTO, and KSTO
- Implementation in fully documented open source **software prototype**
- Practical evaluation on a case study and randomized systems

Future work

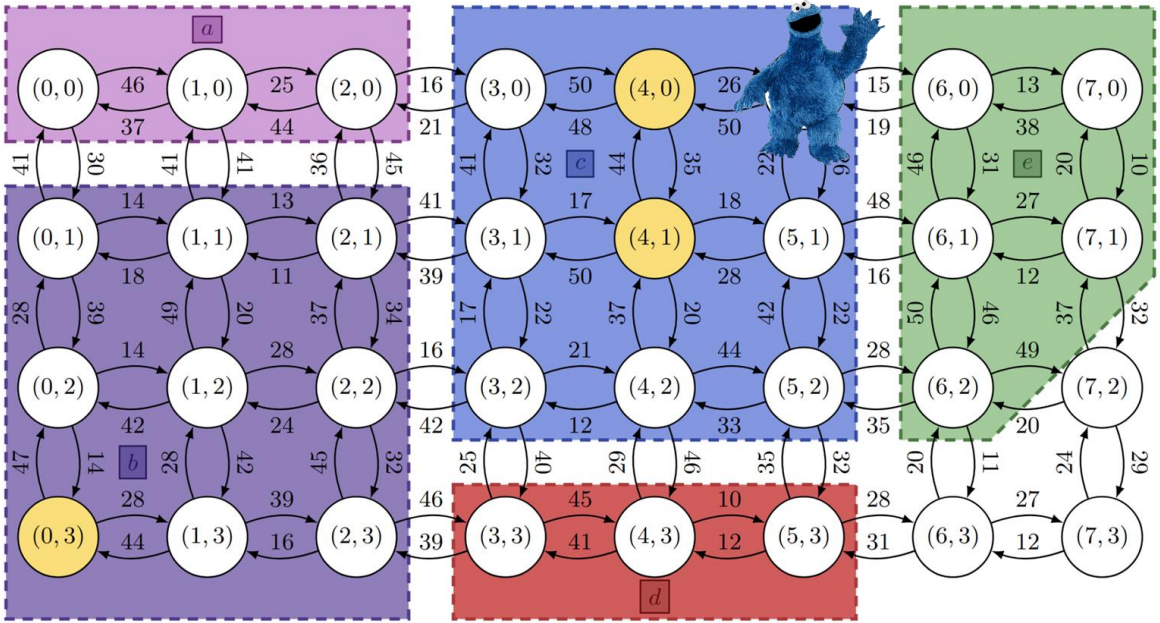
- Opacity **enforcement method**, based on TCC
- Extension to settings with **multiple intruders**
- Extension to **other** opacity notions
- Testing on more **real-world applications**

Case Study - Sensor Network



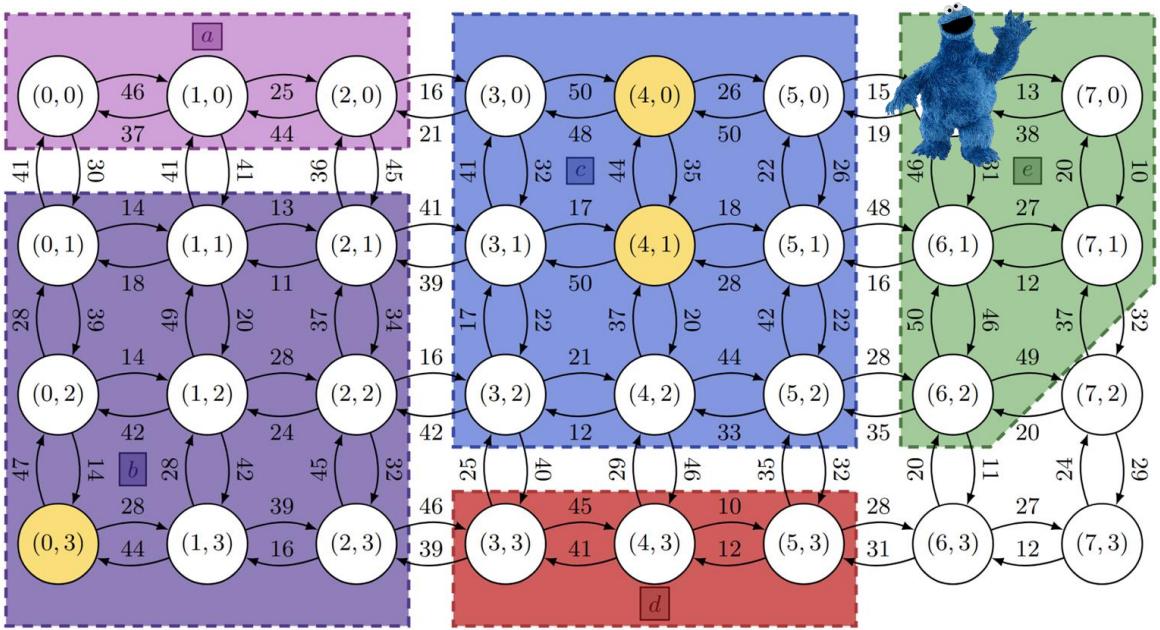
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



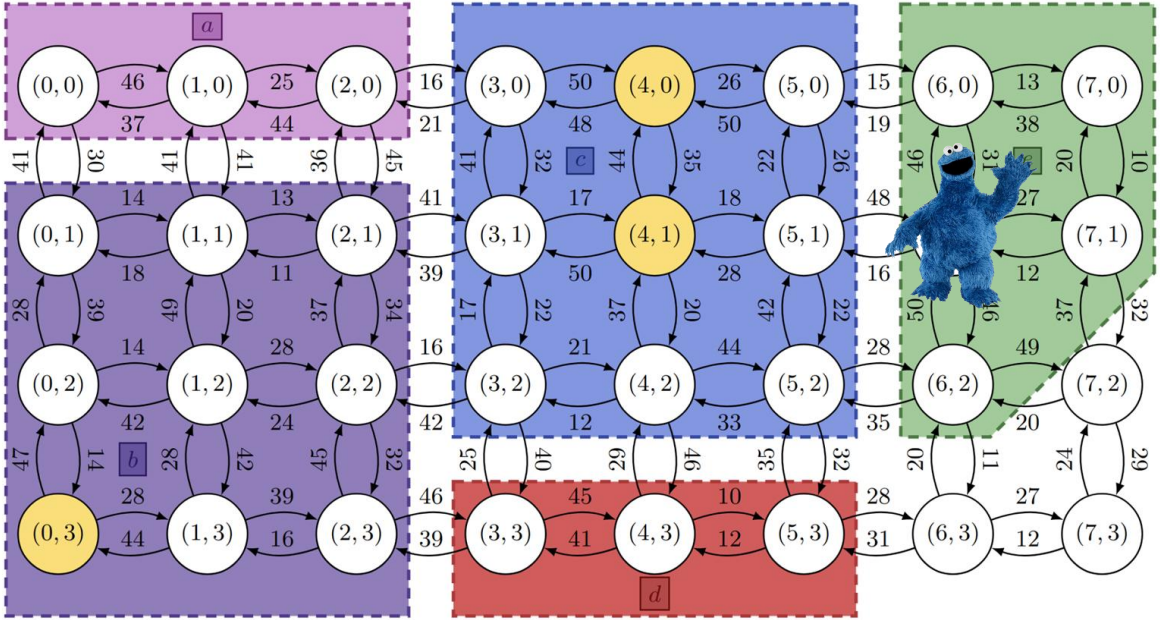
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



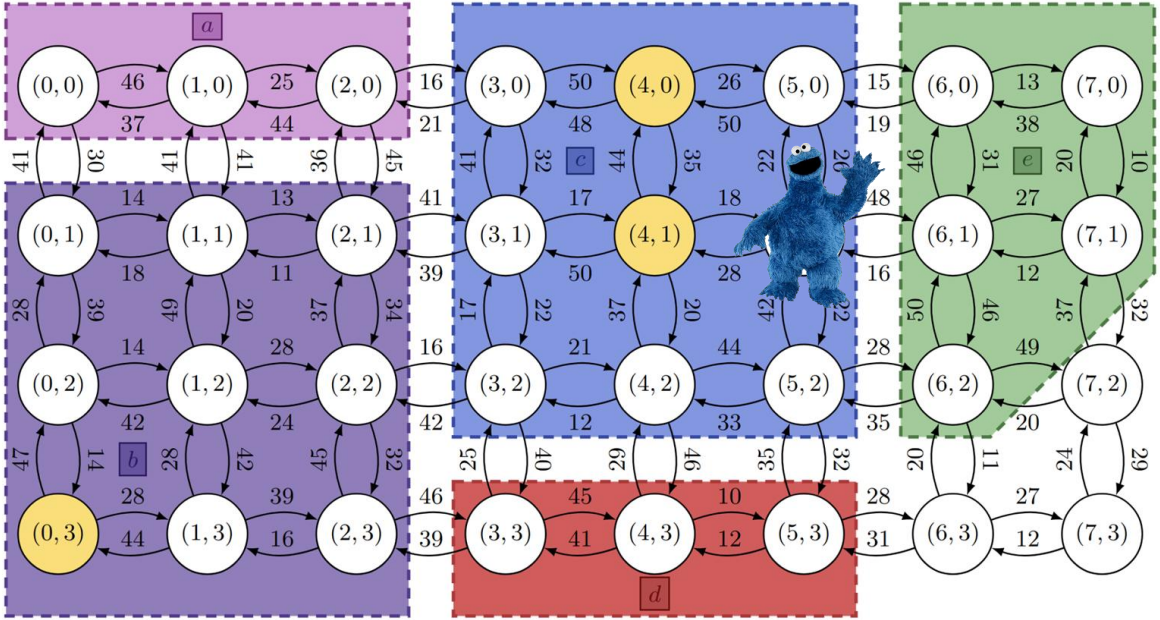
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



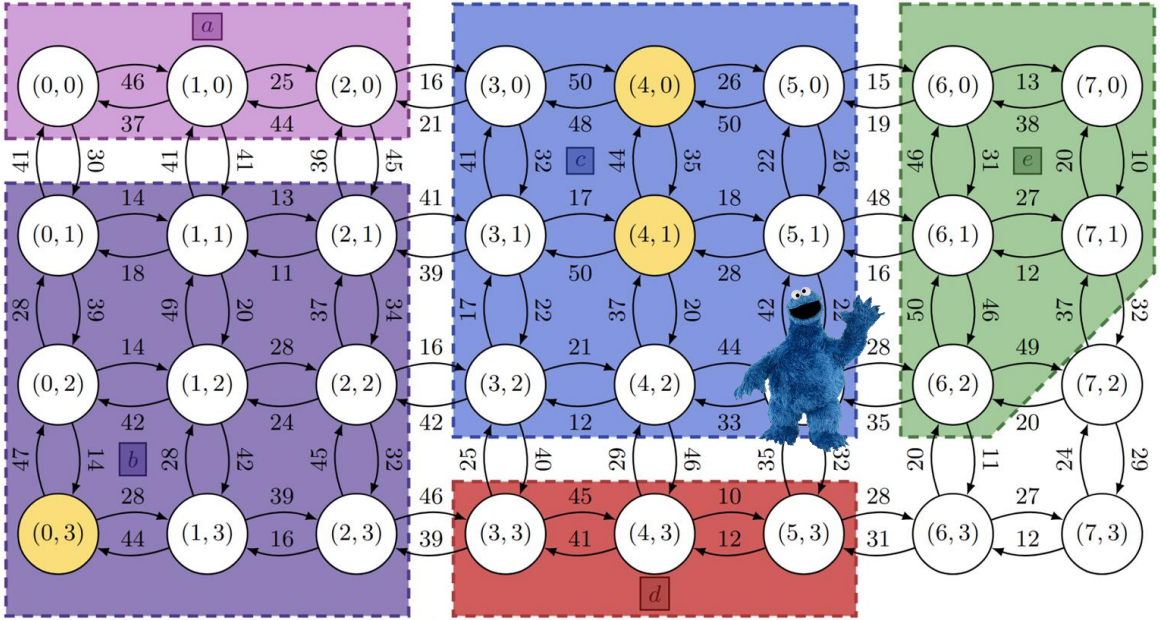
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



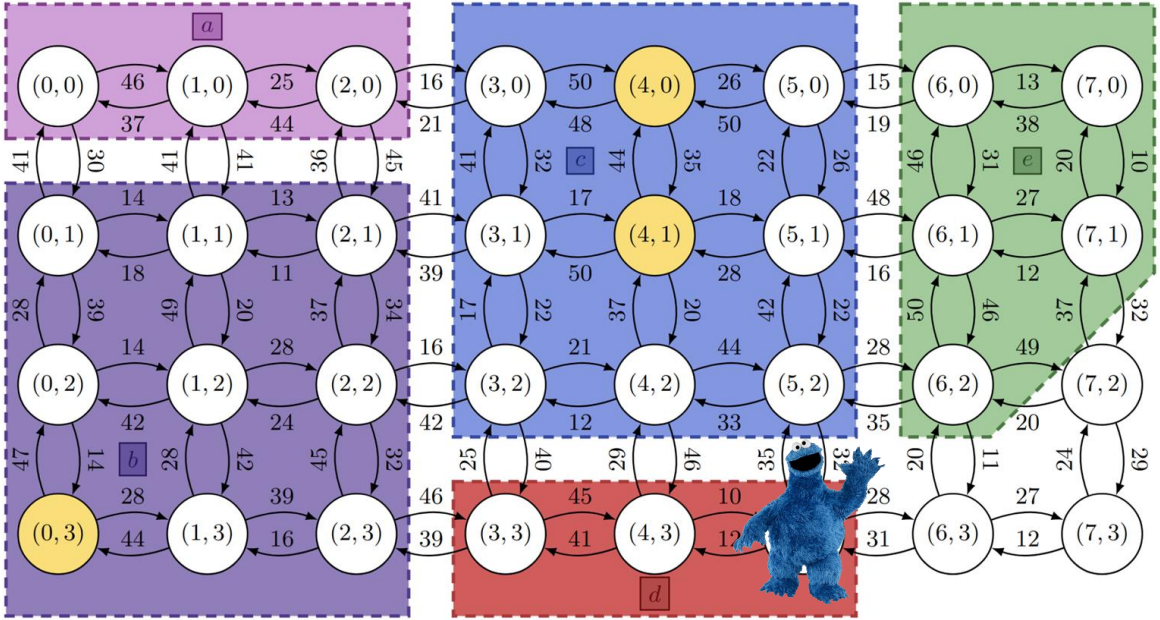
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



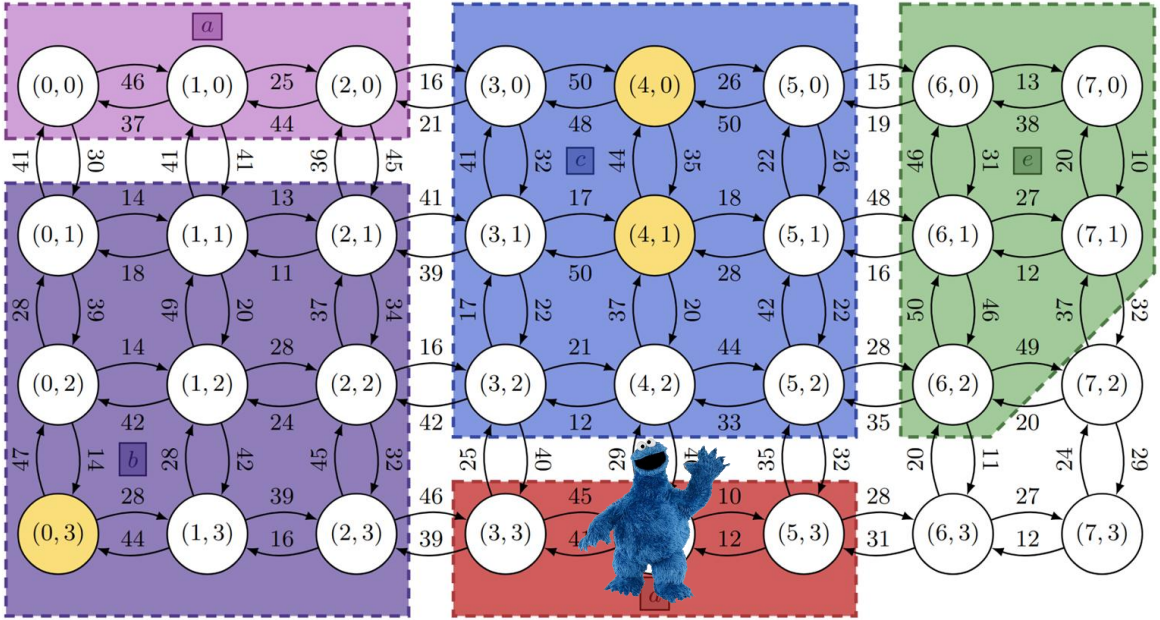
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



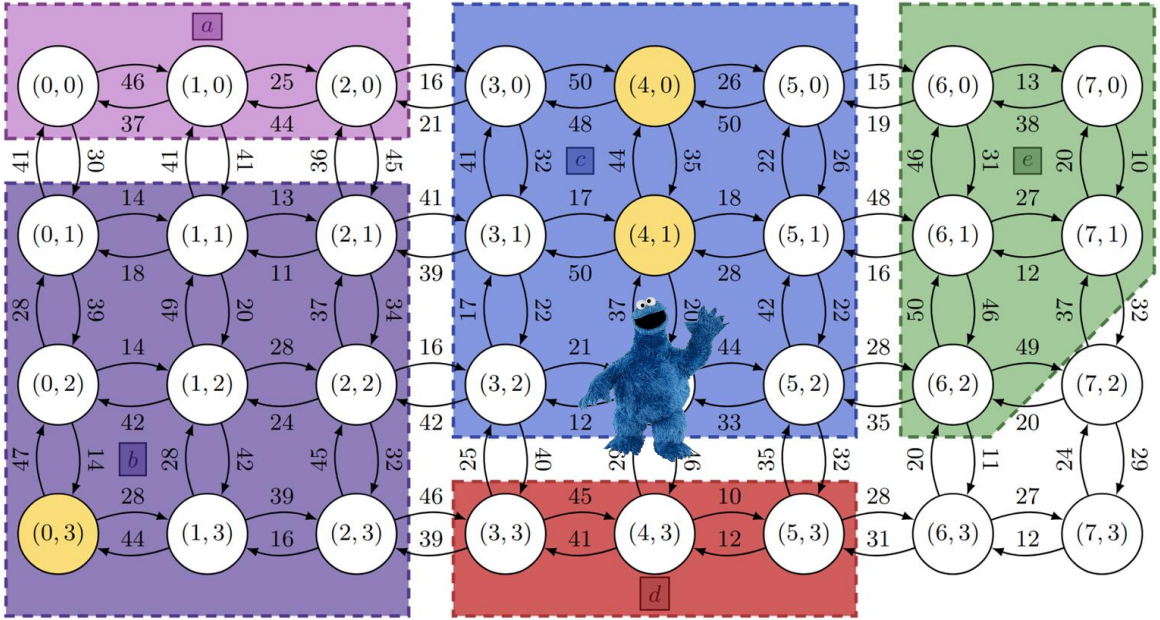
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



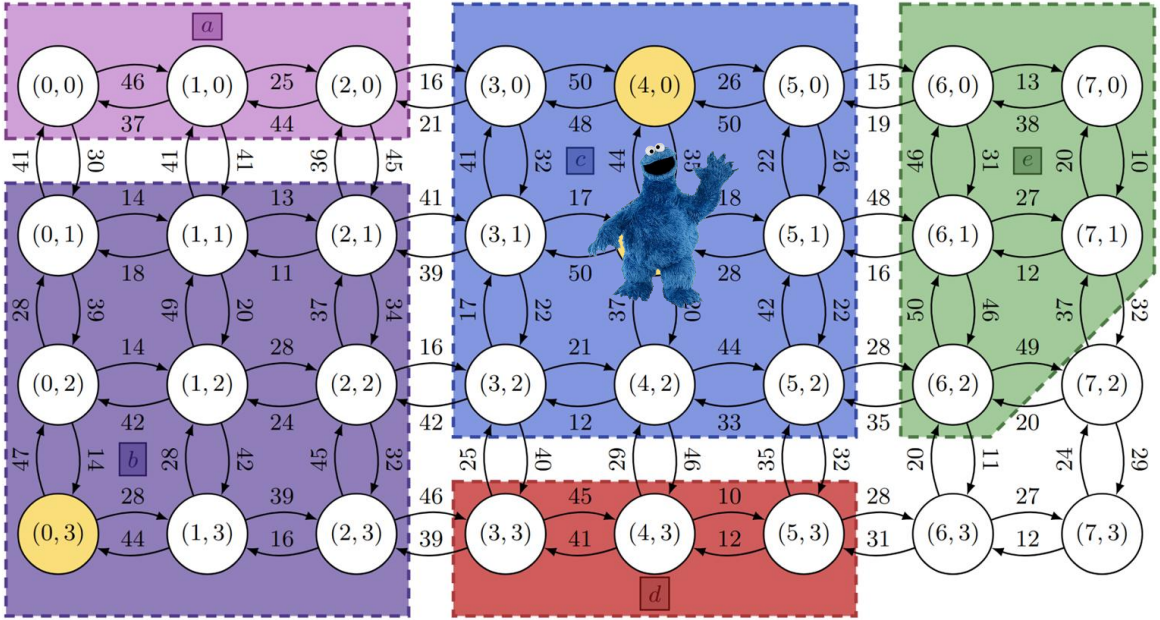
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



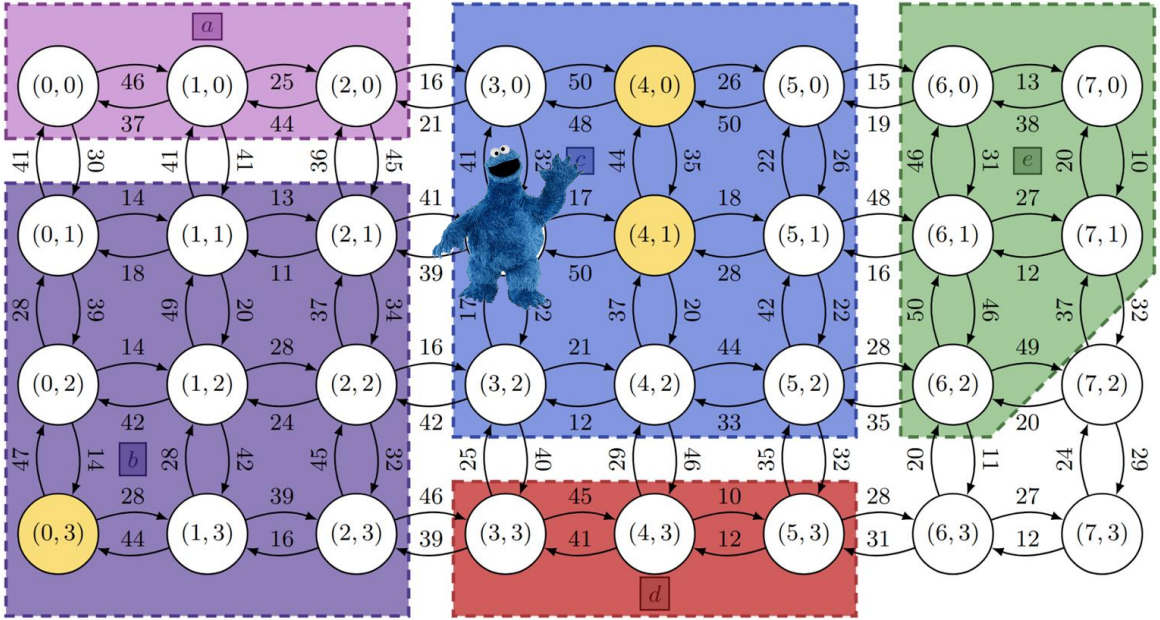
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



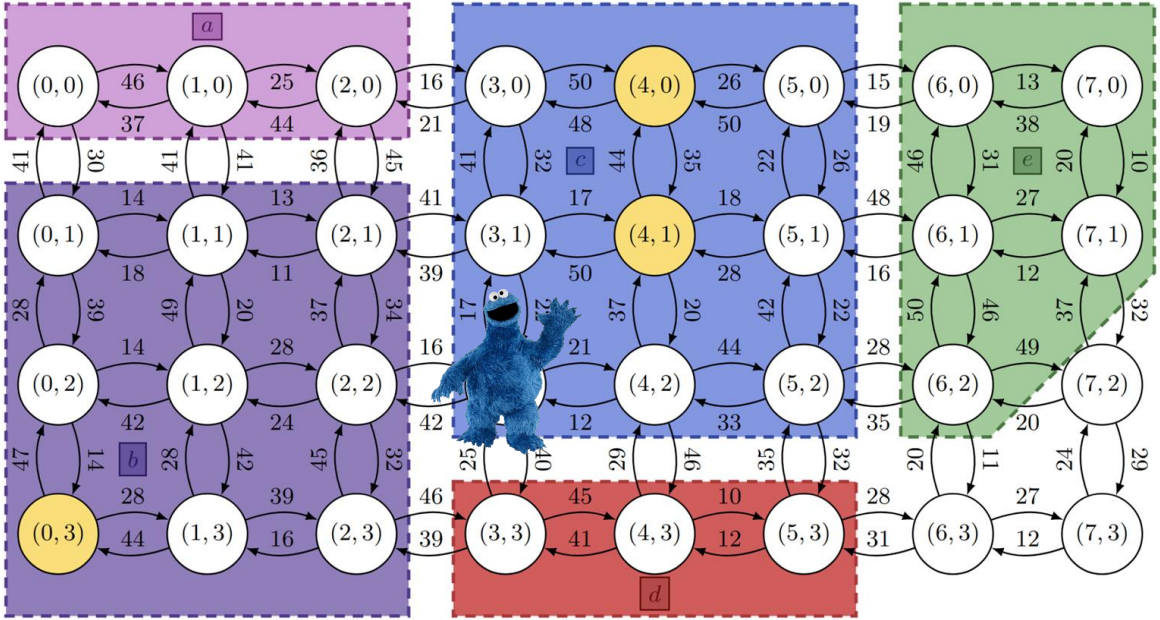
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



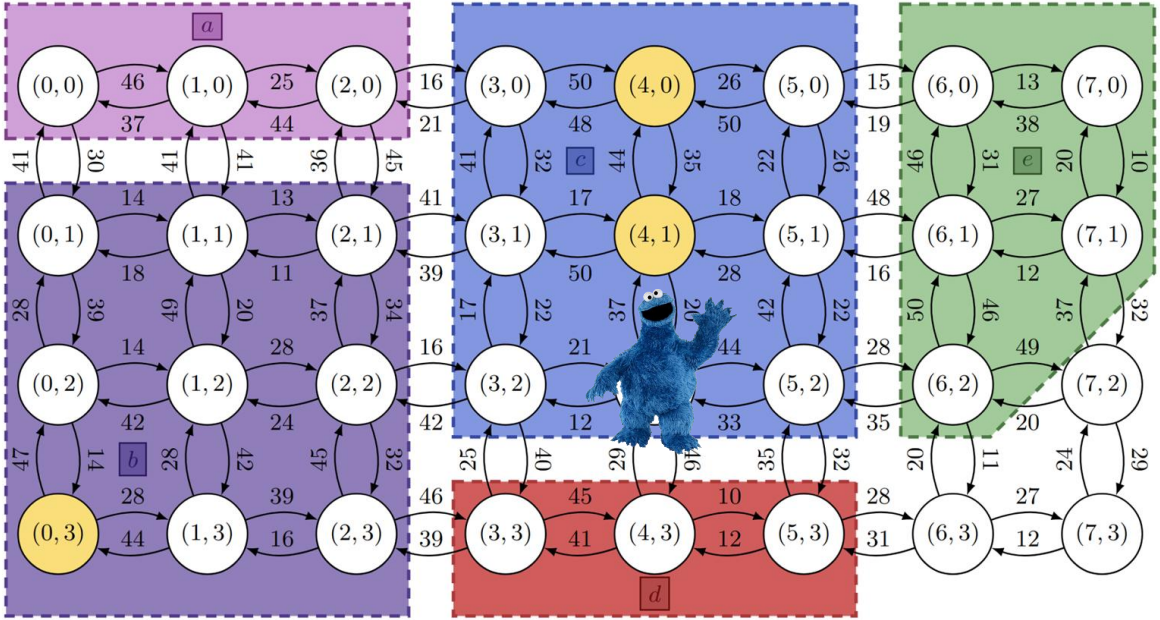
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



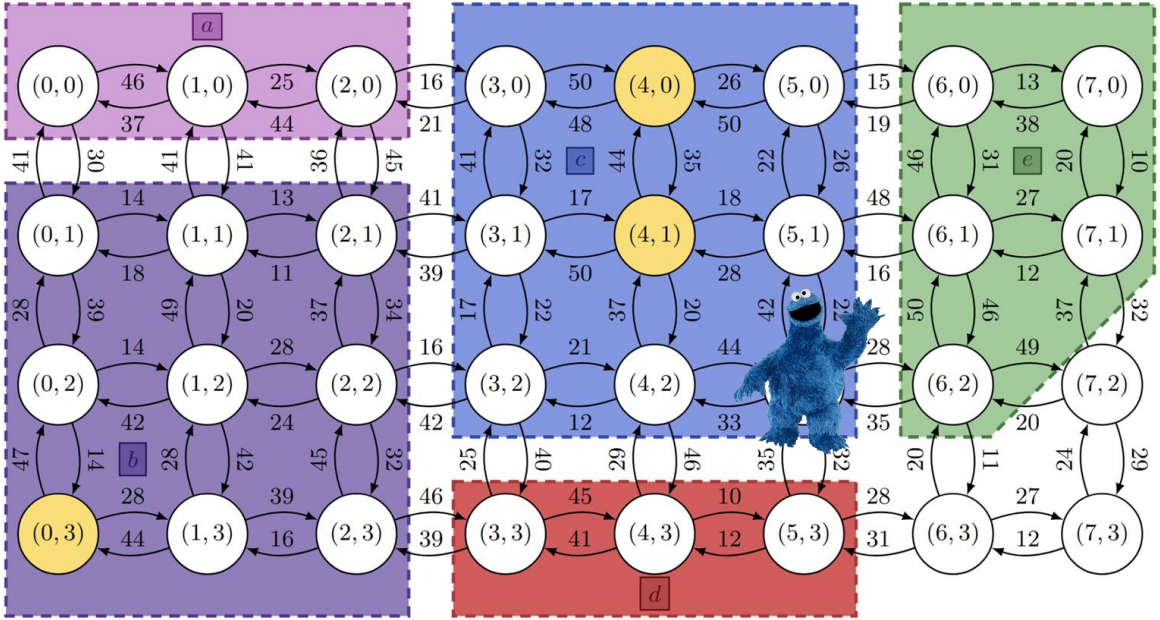
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



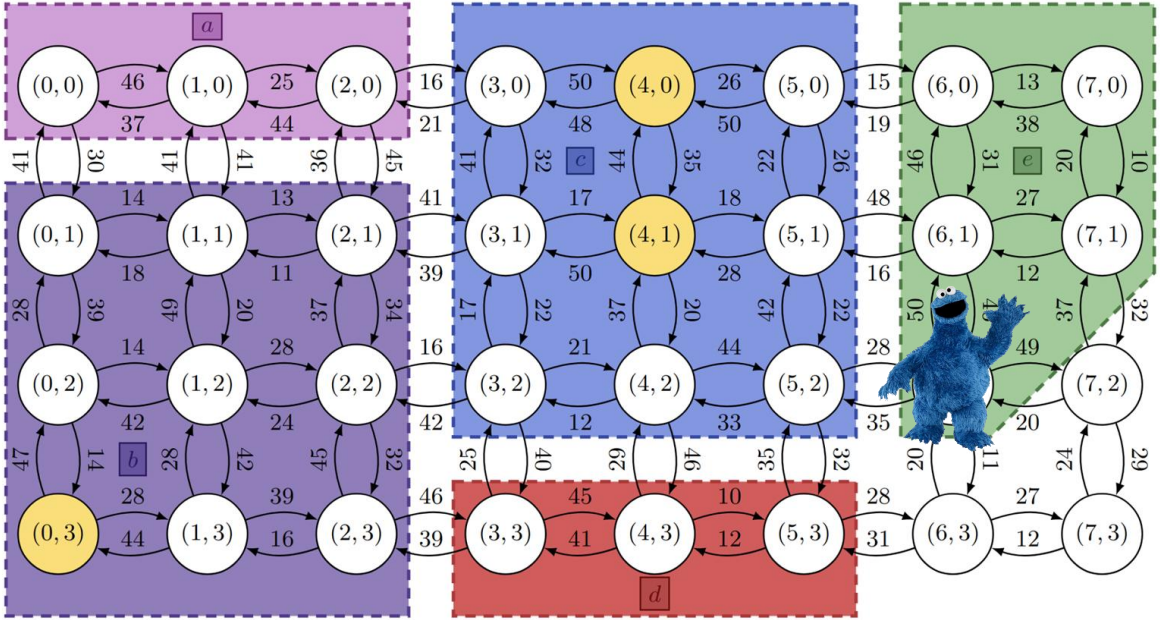
- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



- All states at the **border** are **initial**
- All **yellow** states are **secret**

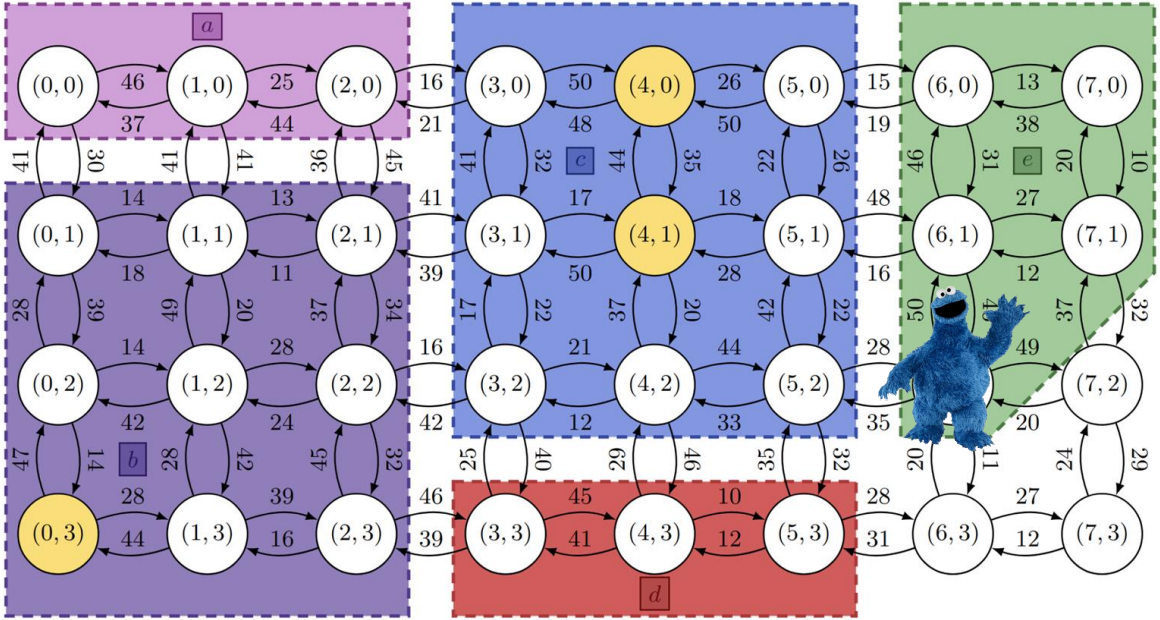
Case Study - Sensor Network



ILTO	✗
CC	3min 6.66s
TCC	35.61s

- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network

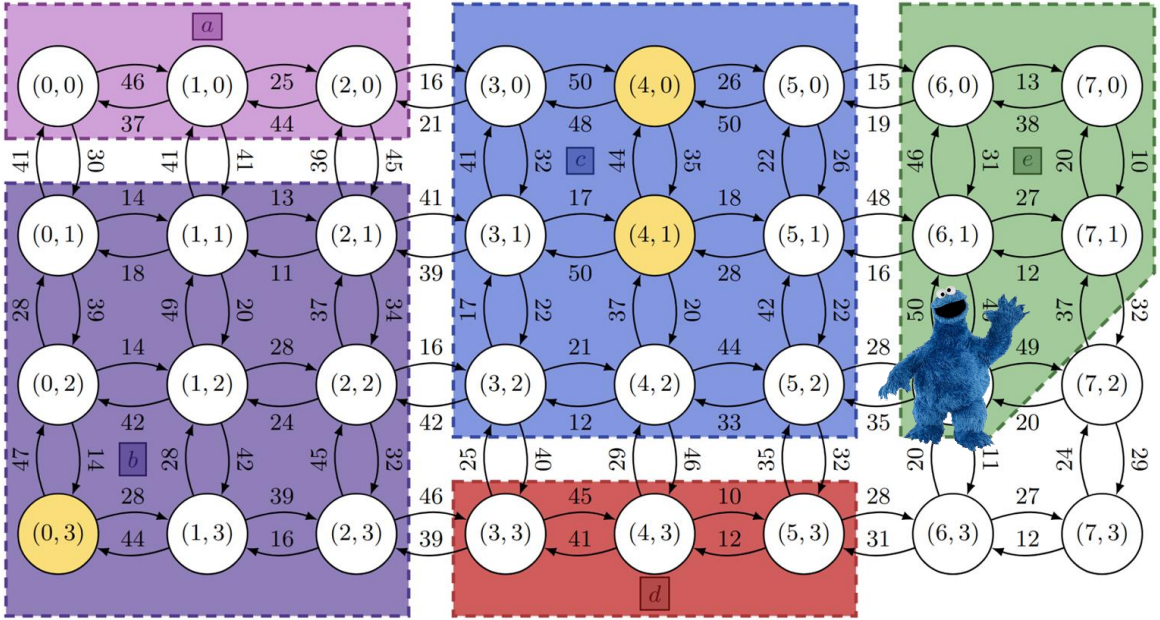


ILTO	✗
CC	3min 6.66s
TCC	35.61s

CLTO	✓
CC	1min 48.23s
TCC	29.76s

- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



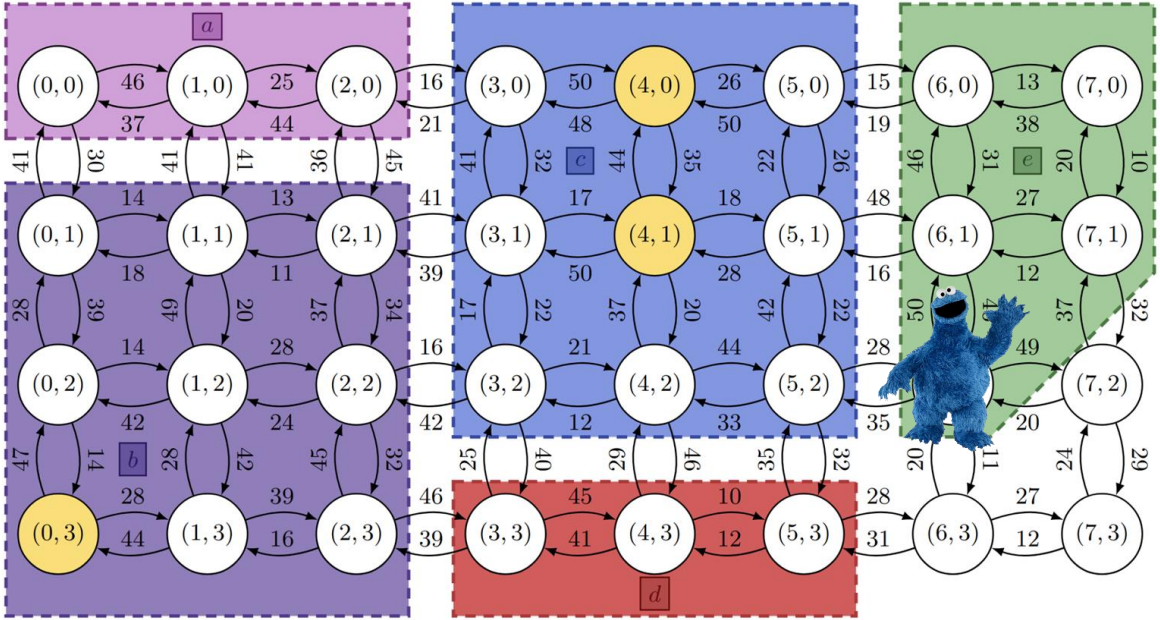
ILTO	✗
CC	3min 6.66s
TCC	35.61s

CLTO	✓
CC	1min 48.23s
TCC	29.76s

ISTO	✗
CC	3min 6.25s
TCC	43.20s

- All states at the **border** are **initial**
- All **yellow** states are **secret**

Case Study - Sensor Network



ILTO	✗
CC	3min 6.66s
TCC	35.61s

CLTO	✓
CC	1min 48.23s
TCC	29.76s

ISTO	✗
CC	3min 6.25s
TCC	43.20s

KSTO (K > 0)	✗
CC	3min 8.67s
TCC	35.58s

- All states at the **border** are **initial**
- All **yellow** states are **secret**