A Polynomial-Time Checkable Sufficient Condition for Deadlock-Freedom of Component-Based Systems

Christoph Minnameier
joint work with Mila Majster-Cederbaum & Moritz Martens

Institute for Computer Science, University of Mannheim, Germany

January 21, 2007

The Setting

- We build on a model for component based systems presented in [Goessler and Sifakis, Component-based Construction of deadlock-free Systems. In FSTTCS, LNCS 2914, 2003.]

The Setting

- We build on a model for component based systems presented in [Goessler and Sifakis, Component-based Construction of deadlock-free Systems. In FSTTCS, LNCS 2914, 2003.]
- Deadlock-Detection in Component-Based Systems is NP-hard [C. Minnameier. Submitted for publication in *IPL*.]

The Setting

- We build on a model for component based systems presented in [Goessler and Sifakis, Component-based Construction of deadlock-free Systems. In FSTTCS, LNCS 2914, 2003.]
- Deadlock-Detection in Component-Based Systems is NP-hard [C. Minnameier. Submitted for publication in *IPL*.]
- We give a polynomial-time computable sufficient condition for deadlock-freedom.

Part 1:
Interaction Systems & Deadlocks

An Interaction System is a Tuple $Sys = (K, \{A_i\}_{i \in K}, C, \{T_i\}_{i \in K})$

- The set of *components* $K = \{1, \dots, n\}$

An Interaction System is a Tuple $Sys = (K, \{A_i\}_{i \in K}, C, \{T_i\}_{i \in K})$

- The set of *components* $K = \{1, \ldots, n\}$
- The sets of *ports* or *actions* $\{A_i\}_{i \in K}$ of a component
  The port sets are pairwise disjoint.

An Interaction System is a Tuple $Sys = (K, \{A_i\}_{i \in K}, C, \{T_i\}_{i \in K})$

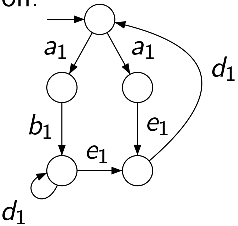
- The set of *components* $K = \{1, \ldots, n\}$
- The sets of *ports* or *actions* $\{A_i\}_{i \in K}$ of a component
  The port sets are pairwise disjoint.
- The set of *connectors* $C = \{c_1, \ldots, c_m\}$
  Connectors are sets of actions. A component can participate in a connector with at most one action.
  Every action of every component has to occur in at least one connector.
  Connectors are maximal w.r.t. set inclusion.

An Interaction System is a Tuple $Sys = (K, \{A_i\}_{i \in K}, C, \{T_i\}_{i \in K})$

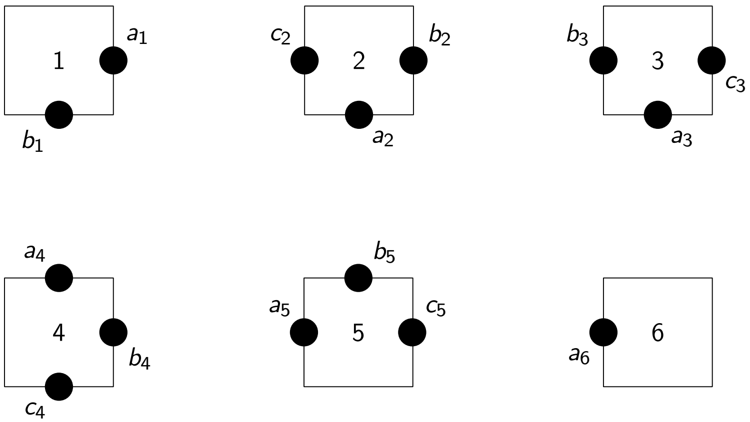
- The set of *components* $K = \{1, \ldots, n\}$
- The sets of *ports* or *actions* $\{A_i\}_{i \in K}$ of a component
  The port sets are pairwise disjoint.
- The set of *connectors* $C = \{c_1, \ldots, c_m\}$
  Connectors are sets of actions. A component can participate in a connector with at most one action.
  Every action of every component has to occur in at least one connector.
  Connectors are maximal w.r.t. set inclusion.

An Interaction System is a Tuple $Sys = (K, \{A_i\}_{i \in K}, C, \{T_i\}_{i \in K})$

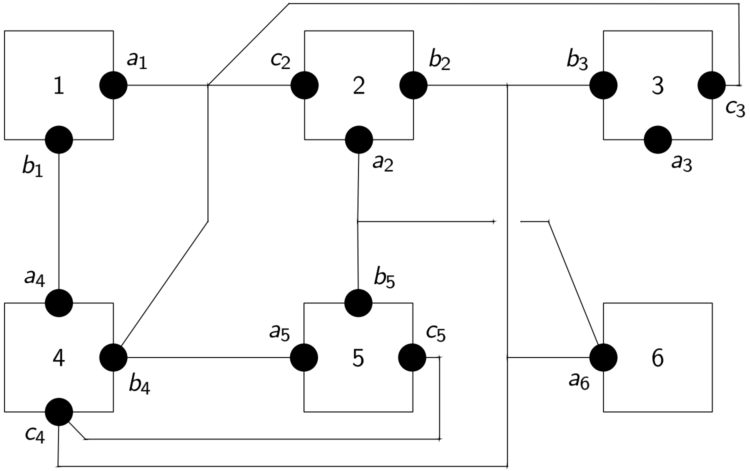
- The set of *components* $K = \{1, \ldots, n\}$
- The sets of *ports* or *actions* $\{A_i\}_{i \in K}$ of a component
  The port sets are pairwise disjoint.
- The set of *connectors* $C = \{c_1, \ldots, c_m\}$
  Connectors are sets of actions. A component can participate in a connector with at most one action.
  Every action of every component has to occur in at least one connector.
  Connectors are maximal w.r.t. set inclusion.
- The local (labeled) transition systems $\{T_i\}_{i \in K}$
  Every node has at least one outgoing edge.



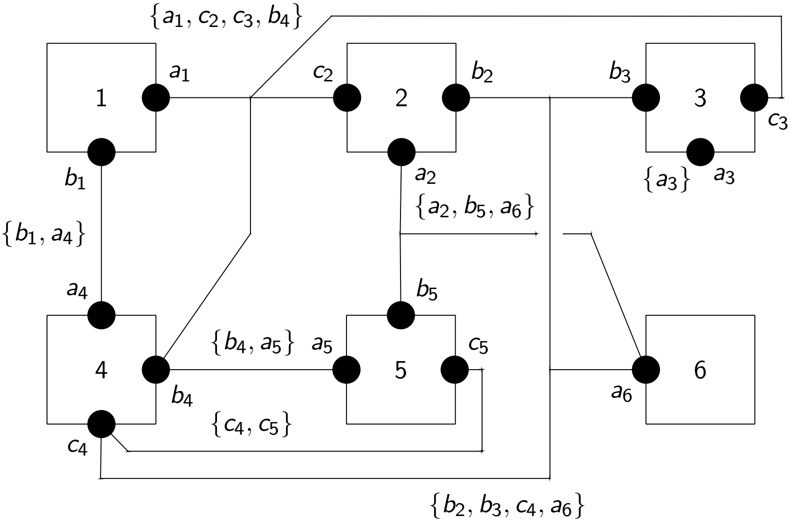
Some Components and their Ports



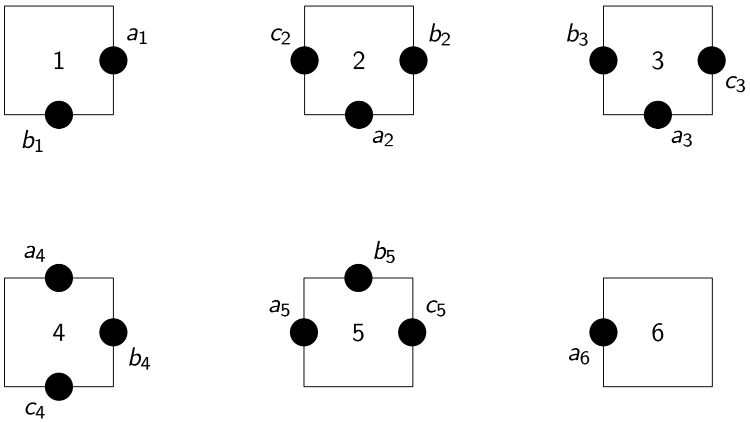
Ports of Components are Connected via Connectors



Ports of Components are Connected via Connectors

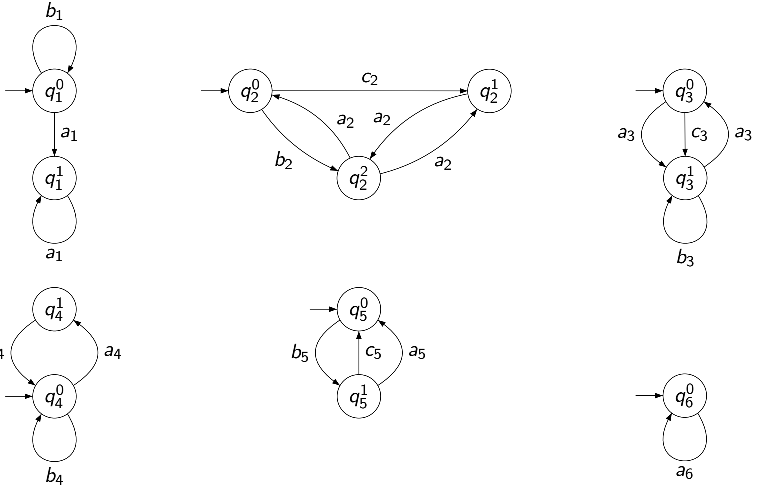


Ports of Components are Connected via Connectors



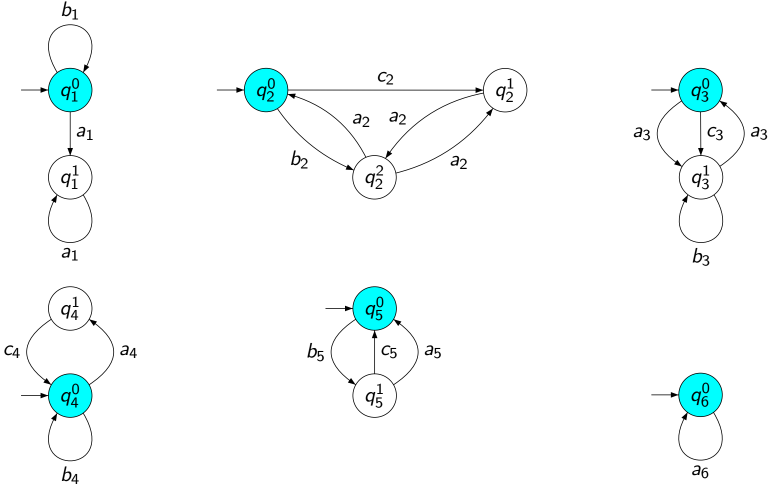
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The Global Behavior of a System



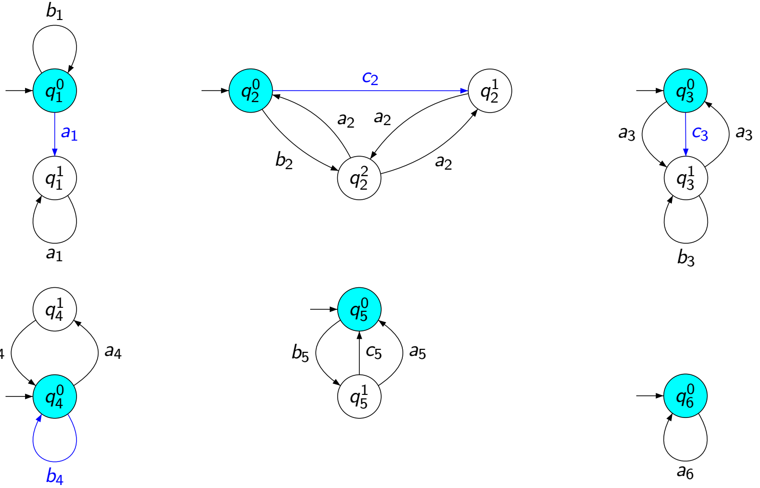
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The Global Behavior of a System



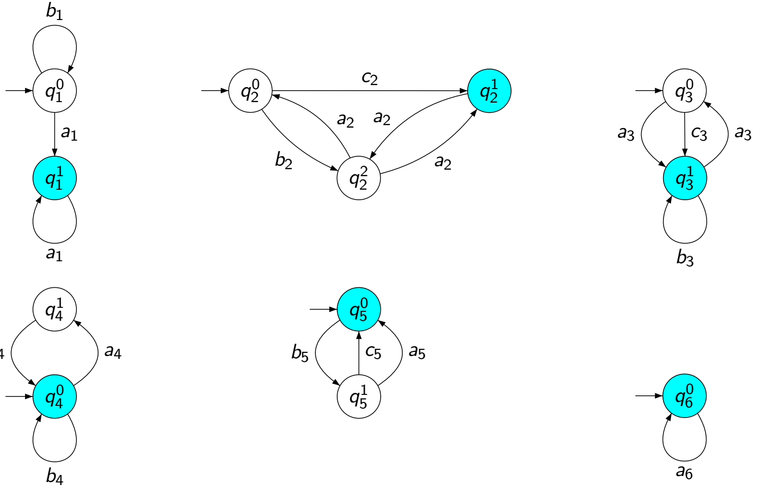
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The Global Behavior of a System



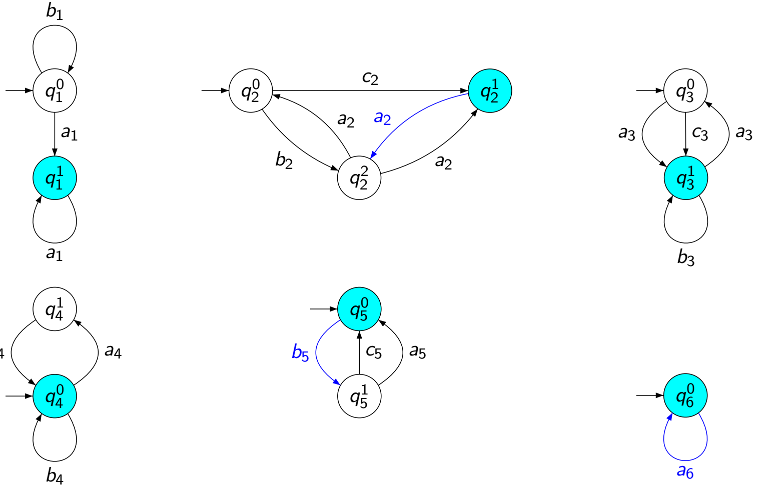
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The Global Behavior of a System



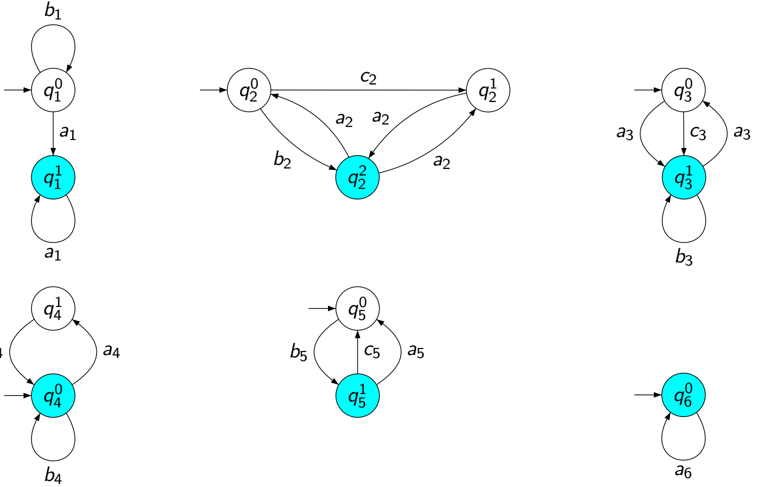
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The Global Behavior of a System



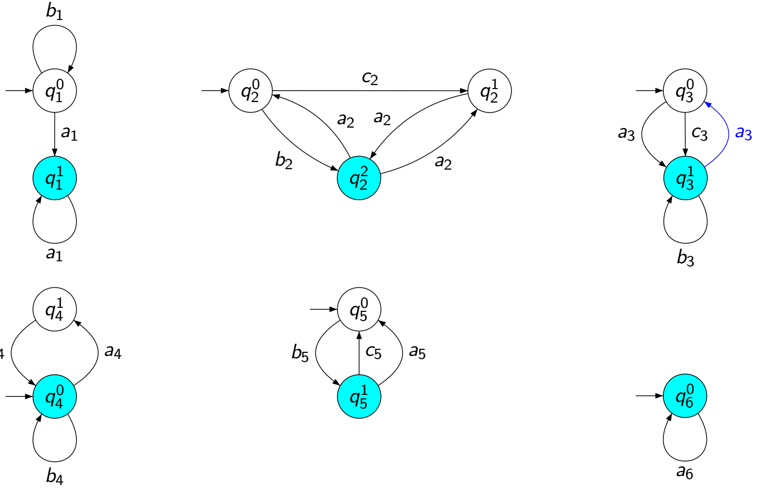
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The Global Behavior of a System



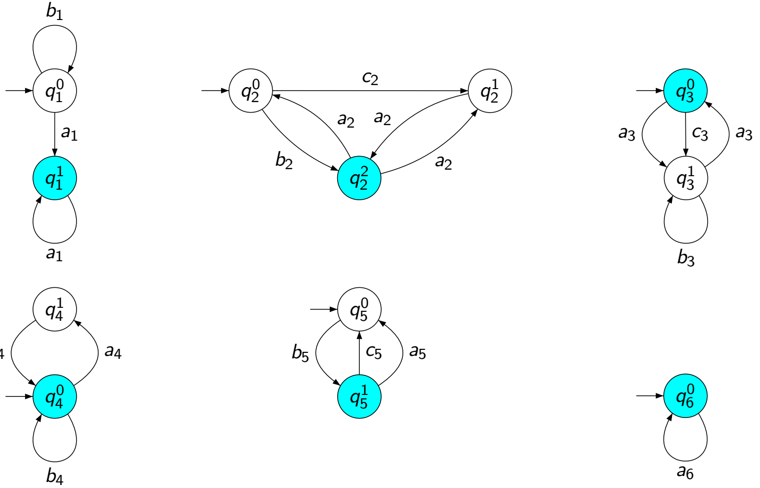
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The System can never be in Global Deadlock



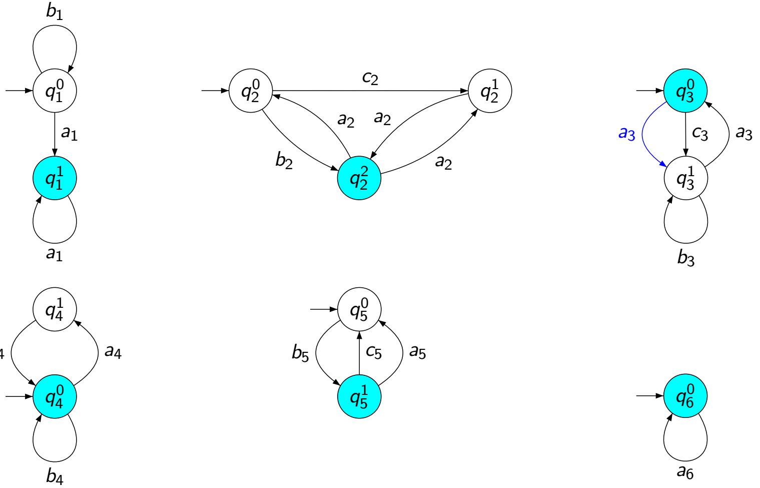
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The System can never be in Global Deadlock



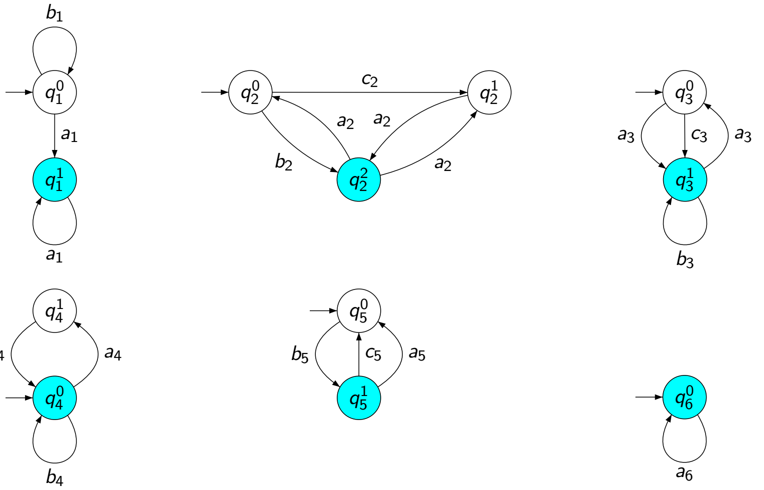
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

The System can never be in Global Deadlock



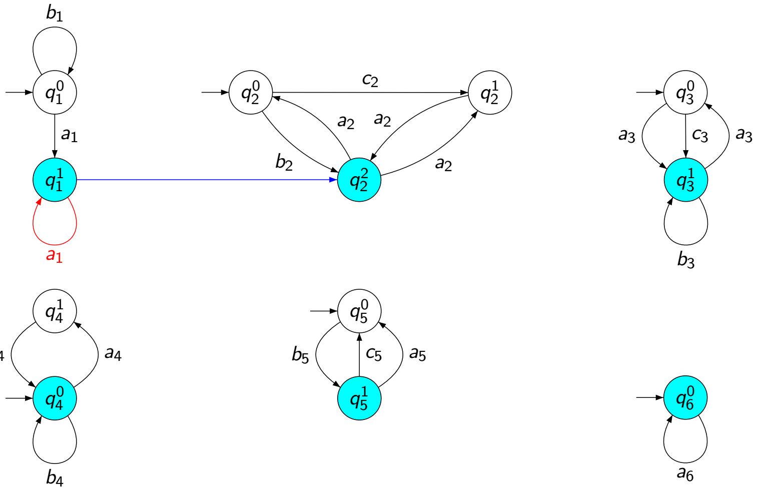
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

But Components $\{1, 2, 4, 5\}$ are in Local Deadlock



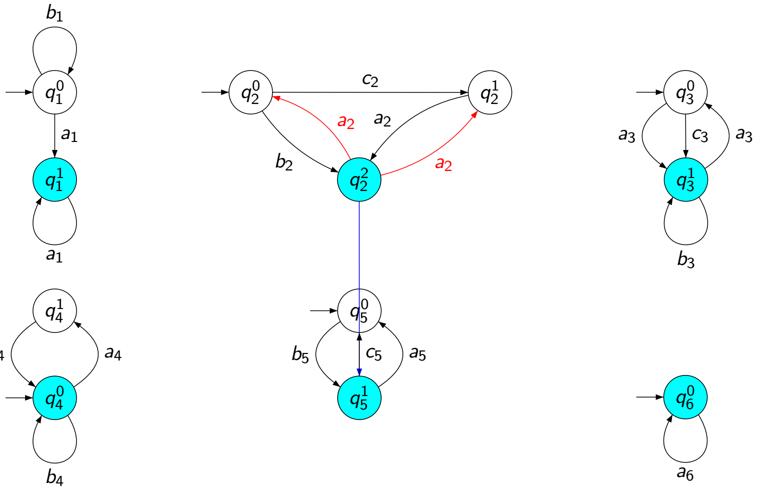
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

But Components $\{1, 2, 4, 5\}$ are in Local Deadlock



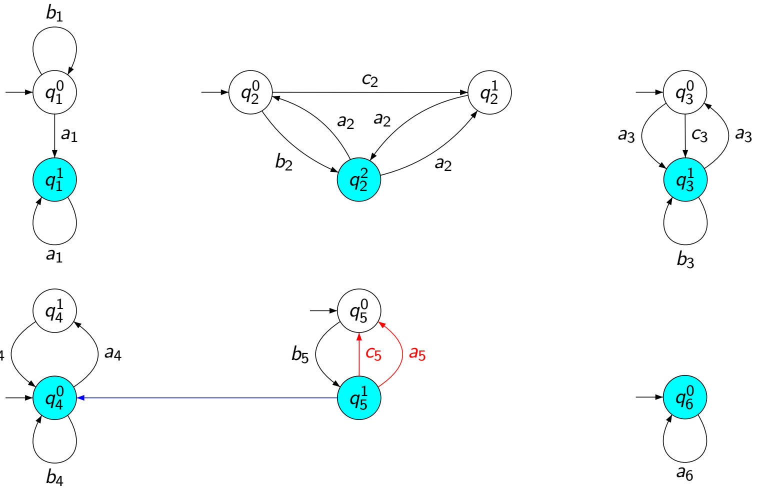
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

But Components $\{1, 2, 4, 5\}$ are in Local Deadlock



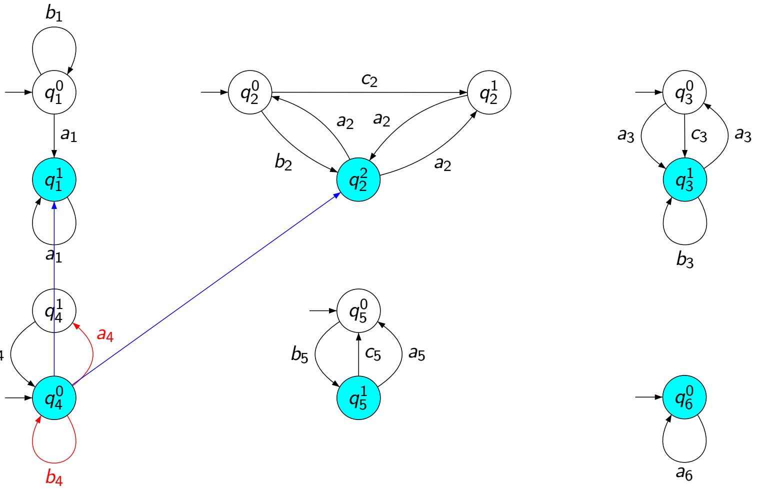
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

But Components $\{1, 2, 4, 5\}$ are in Local Deadlock



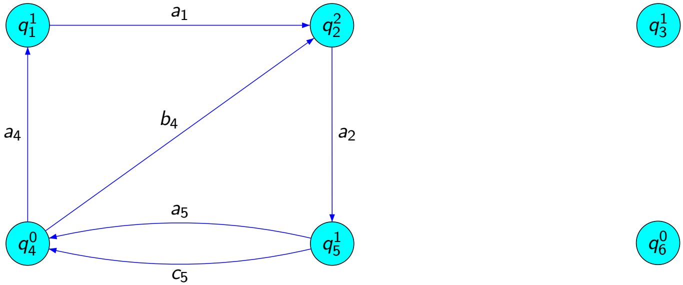
$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

But Components $\{1, 2, 4, 5\}$ are in Local Deadlock



$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

A local Deadlock - Successor-Closed Subgraph



$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

Local and Global Deadlocks

Let $q = (q_1, \ldots, q_n) \in Q$ be a global state.

We say that some non-empty set $D = \{j_1, j_2, \ldots, j_k\} \subseteq K$ of components is in *local deadlock* in q iff

$$\forall i \in D\ \forall c \in C\colon\ c \cap ea(q_i) \neq \emptyset \Rightarrow \exists j \in D\ (c \cap A_j) \not\subseteq ea(q_j)$$

Local and Global Deadlocks

Let $q = (q_1, \ldots, q_n) \in Q$ be a global state.

We say that some non-empty set $D = \{j_1, j_2, \ldots, j_k\} \subseteq K$ of components is in *local deadlock* in q iff

$$\forall i \in D \; \forall c \in C\colon \; c \cap ea(q_i) \neq \emptyset \\ \Rightarrow \exists j \in D \; (c \cap A_j) \not\subseteq ea(q_j)$$

In the example $D = \{1, 2, 4, 5\}$ is in local deadlock.

Local and Global Deadlocks

Let $q = (q_1, \ldots, q_n) \in Q$ be a global state.

We say that some non-empty set $D = \{j_1, j_2, \ldots, j_k\} \subseteq K$ of components is in *local deadlock* in q iff

$$\forall i \in D\ \forall c \in C\colon\ c \cap ea(q_i) \neq \emptyset \\ \Rightarrow \exists j \in D\ (c \cap A_j) \not\subseteq ea(q_j)$$

In the example $D = \{1, 2, 4, 5\}$ is in local deadlock.

We call a local deadlock $D = K$ a *global deadlock*.

Local and Global Deadlocks

Let $q = (q_1, \ldots, q_n) \in Q$ be a global state.

We say that some non-empty set $D = \{j_1, j_2, \ldots, j_k\} \subseteq K$ of components is in *local deadlock* in q iff

$$\forall i \in D\ \forall c \in C\colon\ c \cap ea(q_i) \neq \emptyset \\ \Rightarrow \exists j \in D\ (c \cap A_j) \not\subseteq ea(q_j)$$

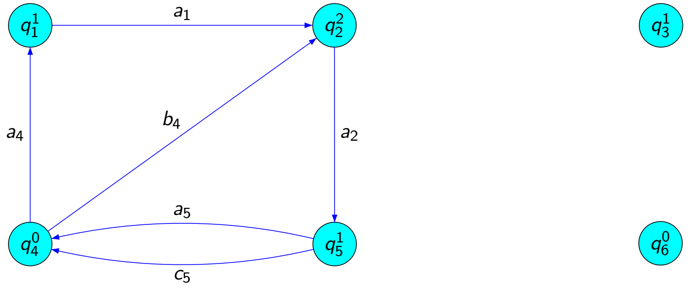
In the example $D = \{1, 2, 4, 5\}$ is in local deadlock.

We call a local deadlock $D = K$ a *global deadlock*.

Deadlock-Detection is NP-hard!

Part 2:
Proving Deadlock-Freedom in Polynomial Time

A Successor-Closed Subgraph implies a Cycle

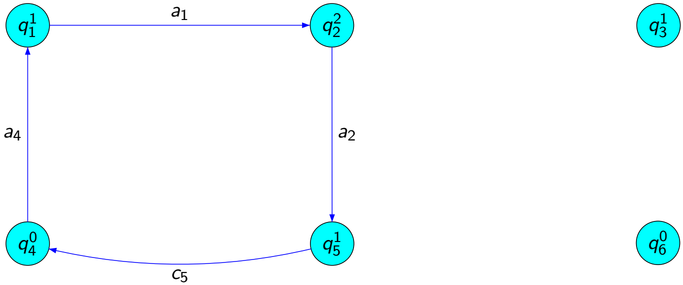


$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

A Successor-Closed Subgraph implies a Cycle

No Cylce (in any reachable global state)

$\Rightarrow$ **No Deadlock (in any reachable global state)**



$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

A Cycle must have been closed somehow

- Assume there is no cycle in the global starting state (easy to check)

A Cycle must have been closed somehow

- Assume there is no cycle in the global starting state (easy to check)
- Let $q^0 \rightarrow \ldots \rightarrow q^D$ a path in the global transition system such that a cycle occurs in q^D for the first time

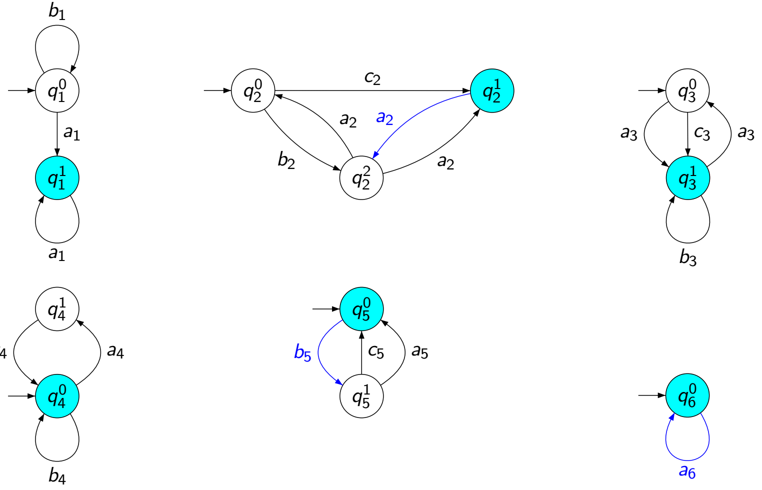
A Cycle must have been closed somehow

- Assume there is no cycle in the global starting state (easy to check)
- Let $q^0 \rightarrow \ldots \rightarrow q^D$ a path in the global transition system such that a cycle occurs in q^D for the first time
- Then there has to be a component (namely one that participates in the cycle) that just (properly) changed its local state

A Cycle must have been closed somehow

- Assume there is no cycle in the global starting state (easy to check)
- Let $q^0 \rightarrow \ldots \rightarrow q^D$ a path in the global transition system such that a cycle occurs in q^D for the first time
- Then there has to be a component (namely one that participates in the cycle) that just (properly) changed its local state
- So there has to be a component that just (properly) changed its local state in such a way that:
  - it waits for some component and
  - it is waited for by some component

The last global Transition before the Cycle occured

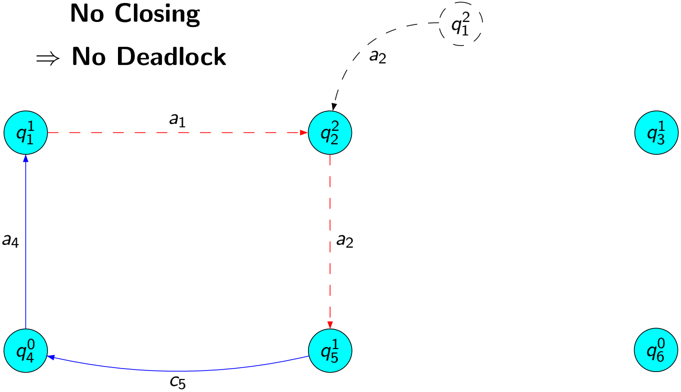


$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

A Cycle implies a Closing

No Closing

$\Rightarrow$ **No Deadlock**

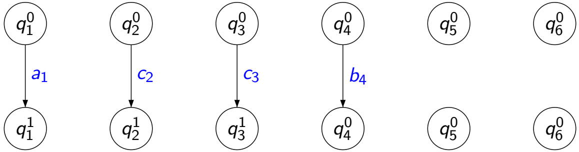


$$C = \{\{a_3\}, \{a_1, c_2, c_3, b_4\}, \{a_2, b_5, a_6\}, \{b_1, a_4\}, \{c_4, a_5\}, \{c_4, c_5\}, \{b_2, b_3, c_4, a_6\}\}$$

Detecting this Closing in the System

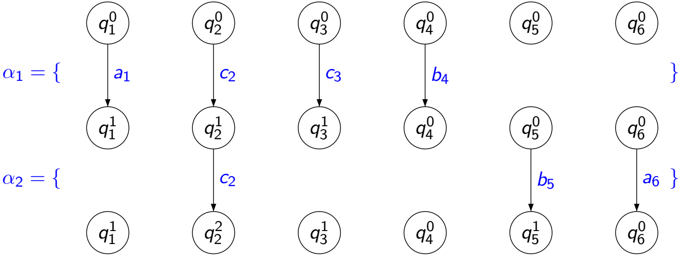
Detecting this Closing in the System

$\alpha_1 = \{$

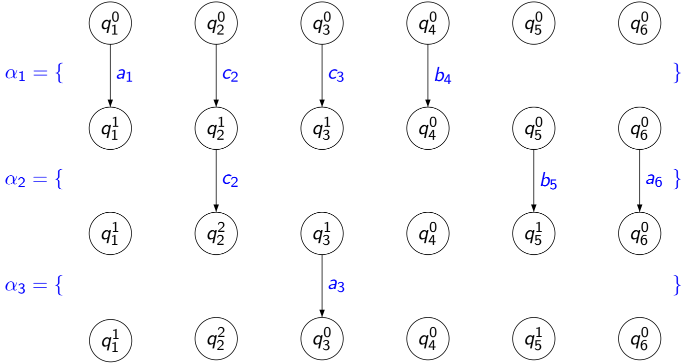


$\}$

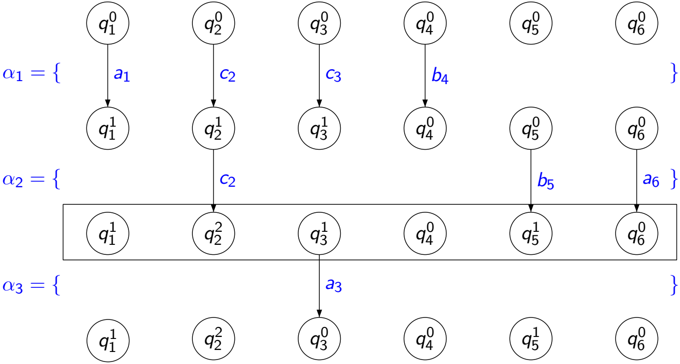
Detecting this Closing in the System



Detecting this Closing in the System

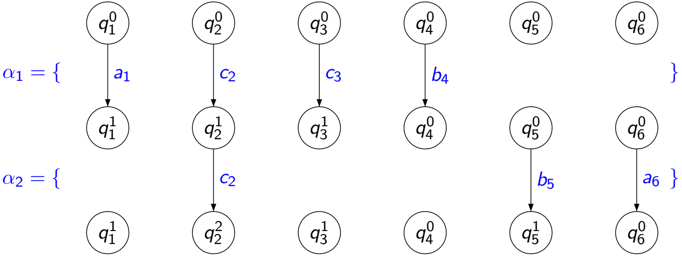


Detecting this Closing in the System

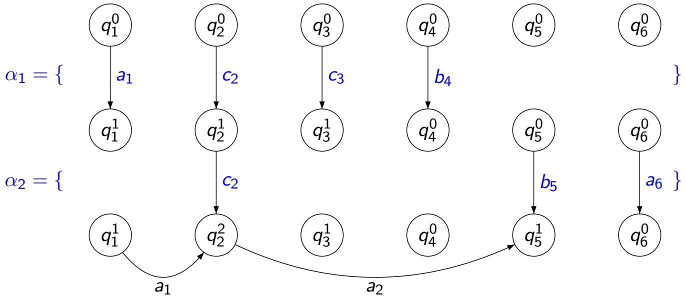


The state where the Cycle occured for the first time.

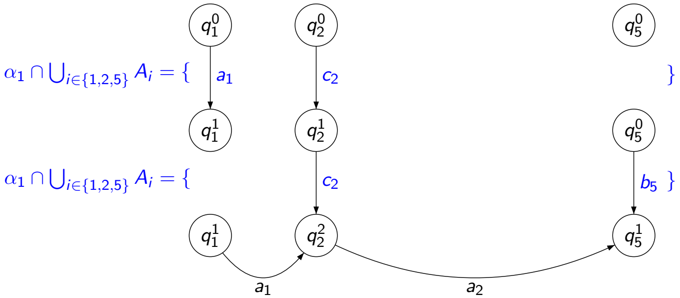
Detecting this Closing in the System



Detecting this Closing in the System



Detecting this Closing in a Subsystem



The witness of the potential formation of a cycle is still present after restricting the connectors to the action sets of the observed components.

Complexity and Parametrization

- The Algorithm performs a reachability analysis for each subsystem consisting of 3 components. The number of such subsystems is in $O(n^3)$.

Complexity and Parametrization

- The Algorithm performs a reachability analysis for each subsystem consisting of 3 components. The number of such subsystems is in $O(n^3)$.
- Each such subsystem has at most m^3 states, where m is the size of a largest local transition system.

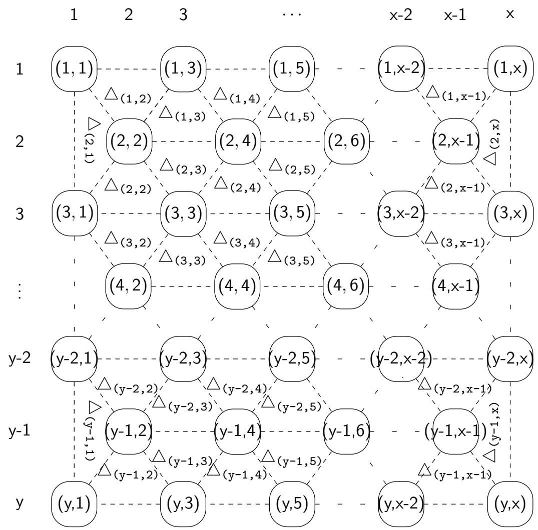
Complexity and Parametrization

- The Algorithm performs a reachability analysis for each subsystem consisting of 3 components. The number of such subsystems is in $O(n^3)$.
- Each such subsystem has at most m^3 states, where m is the size of a largest local transition system.
- To check whether there is a component that performed a proper state change and is now waiting for and waited for takes time $O(|C| \cdot m)$.

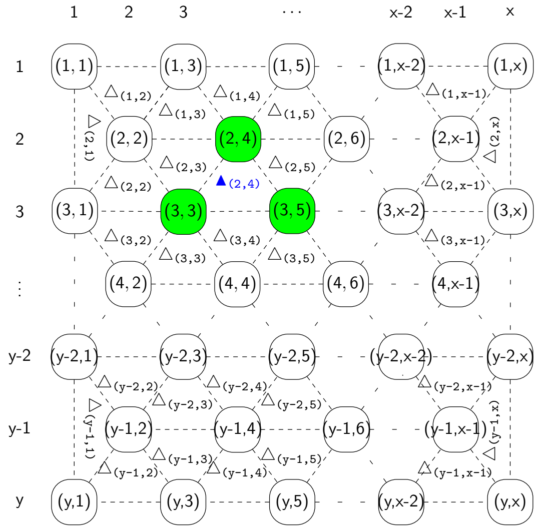
Complexity and Parametrization

- The Algorithm performs a reachability analysis for each subsystem consisting of d components. The number of such subsystems is in $O(n^d)$.
- Each such subsystem has at most m^d states, where m is the size of a largest local transition system.
- To check whether there is a component that performed a proper state change and is now waiting for and waited for takes time $O(|C| \cdot m)$.
- In general, we can observe d components at a time in order to minimize the error in our reachability analyses.

What is it good for? - A Trilateration System

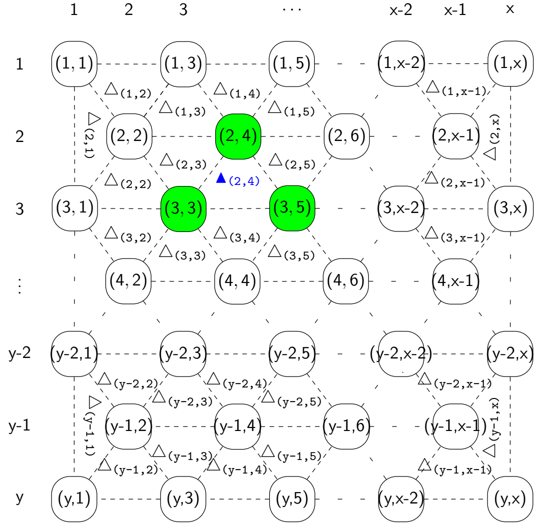


What is it good for? - A Trilateration System



Three components that constitute a triangle may start, perform and end a trilateration cooperation.

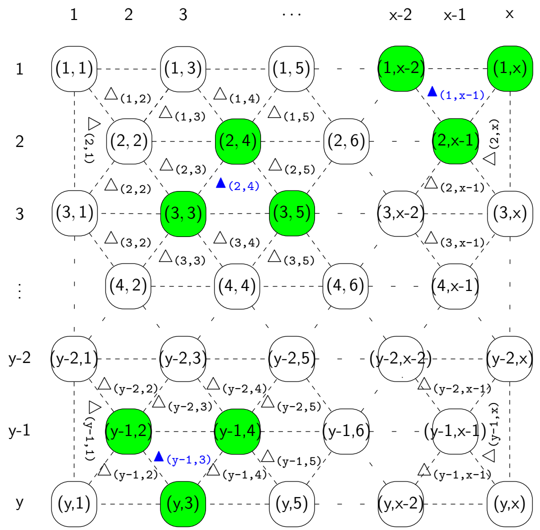
What is it good for? - A Trilateration System



Three components that constitute a triangle may start, perform and end a trilateration cooperation.

Each component may participate in a communication of one of its surrounding triangles at a time.

What is it good for? - A Trilateration System

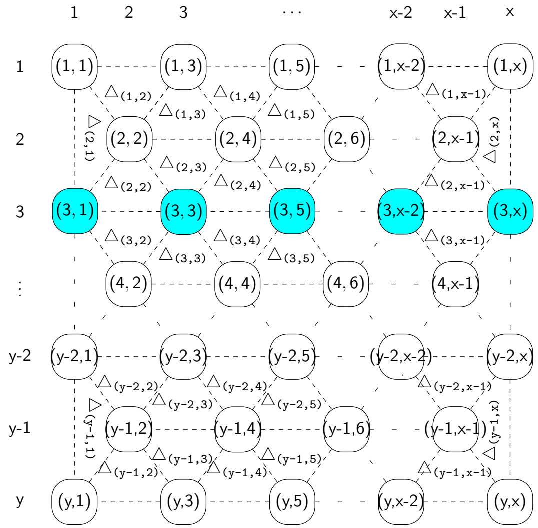


Three components that constitute a triangle may start, perform and end a trilateration cooperation.

Each component may participate in a communication of one of its surrounding triangles at a time.

This yields a reachable global state space whose size is exponential in n.

What is it good for? - A Trilateration System



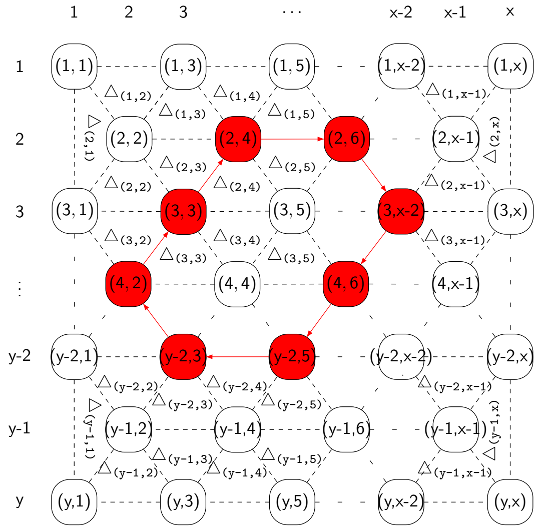
Three components that constitute a triangle may start, perform and end a trilateration cooperation.

Each component may participate in a communication of one of its surrounding triangles at a time.

This yields a reachable global state space whose size is exponential in n.

Each component may also participate in a maintenance-interaction together with the other components in the same row.
⇒ Arbitrarily large connectors.

What is it good for? - A Trilateration System



Three components that constitute a triangle may start, perform and end a trilateration cooperation.

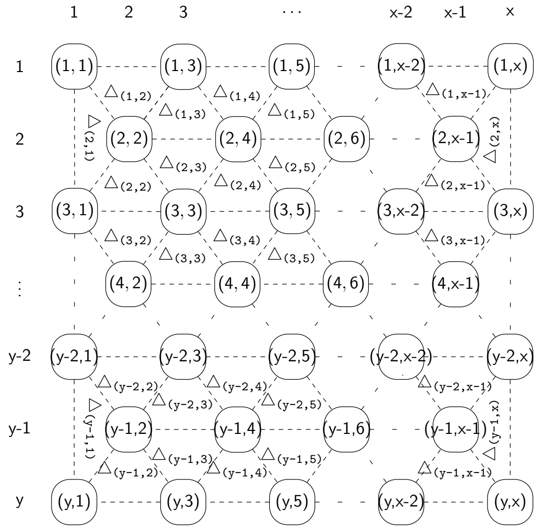
Each component may participate in a communication of one of its surrounding triangles at a time.

This yields a reachable global state space whose size is exponential in n.

Each component may also participate in a maintenance-interaction together with the other components in the same row.
⇒ Arbitrarily large connectors.

There are (unreachable) global states that contain deadlocks.

What is it good for? - A Trilateration System



Three components that constitute a triangle may start, perform and end a trilateration cooperation.

Each component may participate in a communication of one of its surrounding triangles at a time.

This yields a reachable global state space whose size is exponential in n.

Each component may also participate in a maintenance-interaction together with the other components in the same row.
⇒ Arbitrarily large connectors.

There are (unreachable) global states that contain deadlocks.

The system can be proven deadlock-free by observing subsystems of size 3! √

Conclusion

- We introduced a sufficient condition for deadlock-freedom of component-based systems

Conclusion

- We introduced a sufficient condition for deadlock-freedom of component-based systems
- The condition can be checked within subsystems which yields a polynomial time bound

Conclusion

- We introduced a sufficient condition for deadlock-freedom of component-based systems
- The condition can be checked within subsystems which yields a polynomial time bound
- The size of the subsystems serves as a parameter which enables us to do a trade-off between time and accuracy