Verification of Buffered Dynamic Register Automata

Parosh Aziz Abdulla

Mohammed Faouzi Atig

Ahmet Kara

Othmane Rezine

UPPSALA UNIVERSITET

technische universität dortmund

Communication Protocols Analysis

Verification of a class of protocols:

- **Dynamic creation** of processes
  - **Unique** ID for each process
- Finite number of **registers** per process
  - Used to store the ID of others
- **Asynchronous** communication

Dynamic Register Automata

[Bollig et al. 2010, 2013]

- Dynamic Communication Automata
- Asynchronous Communication
- Applications:
  - Leader Election Protocol
  - Peer To Peer Protocol
  - Ad Hoc Networks

Dynamic Register Automata

Verification of Buffered Dynamic Register Automata

[Bollig et al. 2010, 2013]

[Parosh et al. 2014]

- Verification of Dynamic Register Automata
- Rendez-vous Communication
- Sub-Classes for which basic verification properties are decidable

Dynamic Register Automata

[Bollig et al. 2010, 2013]

[Parosh et al. 2014]

[In This Work]

- Verification of Dynamic Register Automata
- Asynchronous Communication
- Sub-Classes for which basic verification properties are decidable

Buffered Dynamic Register Automata

Formal Model

Operational Semantics

Problem

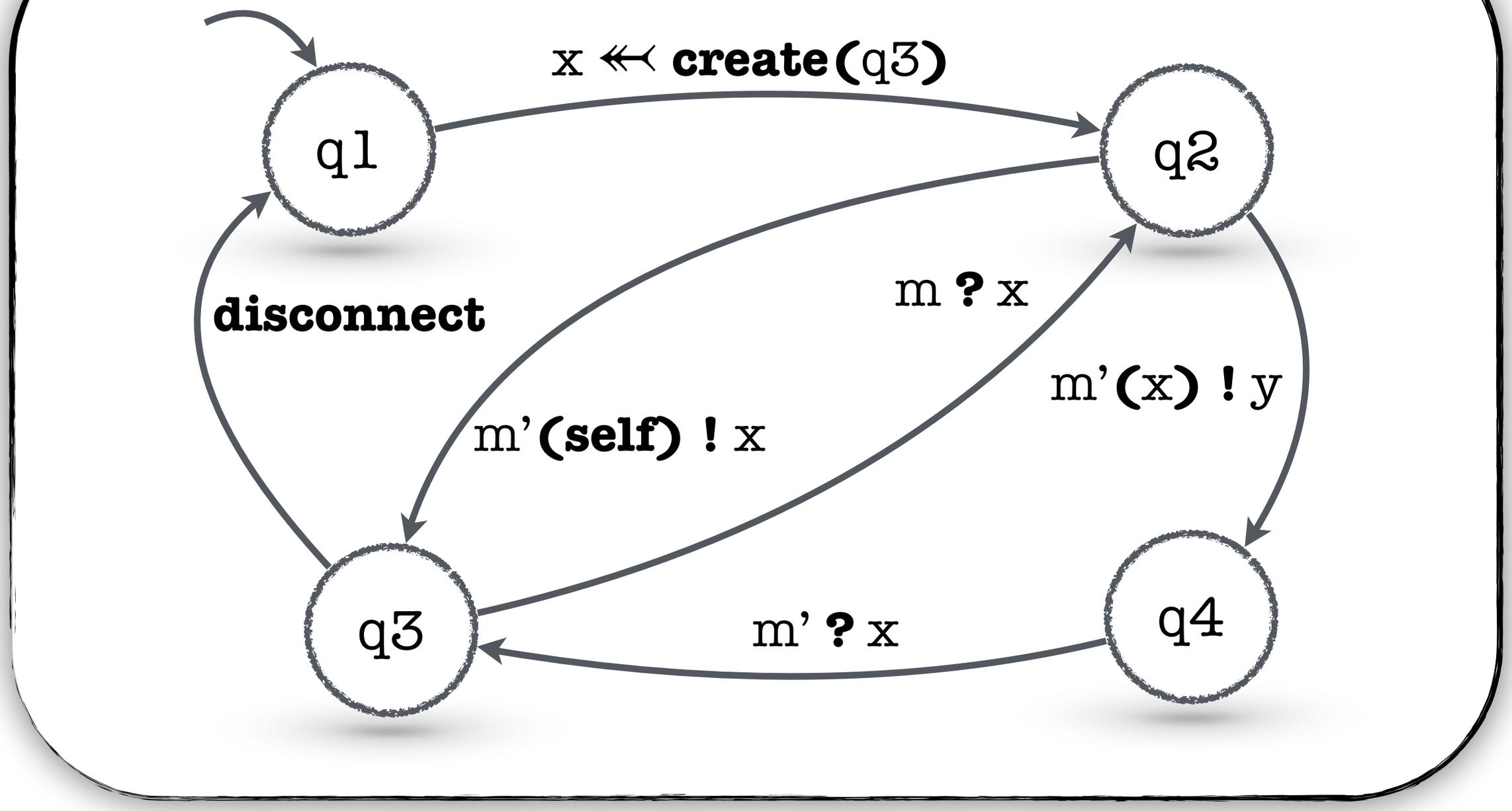
Approach

Results

Formal Model

Verification of Buffered Dynamic Register Automata

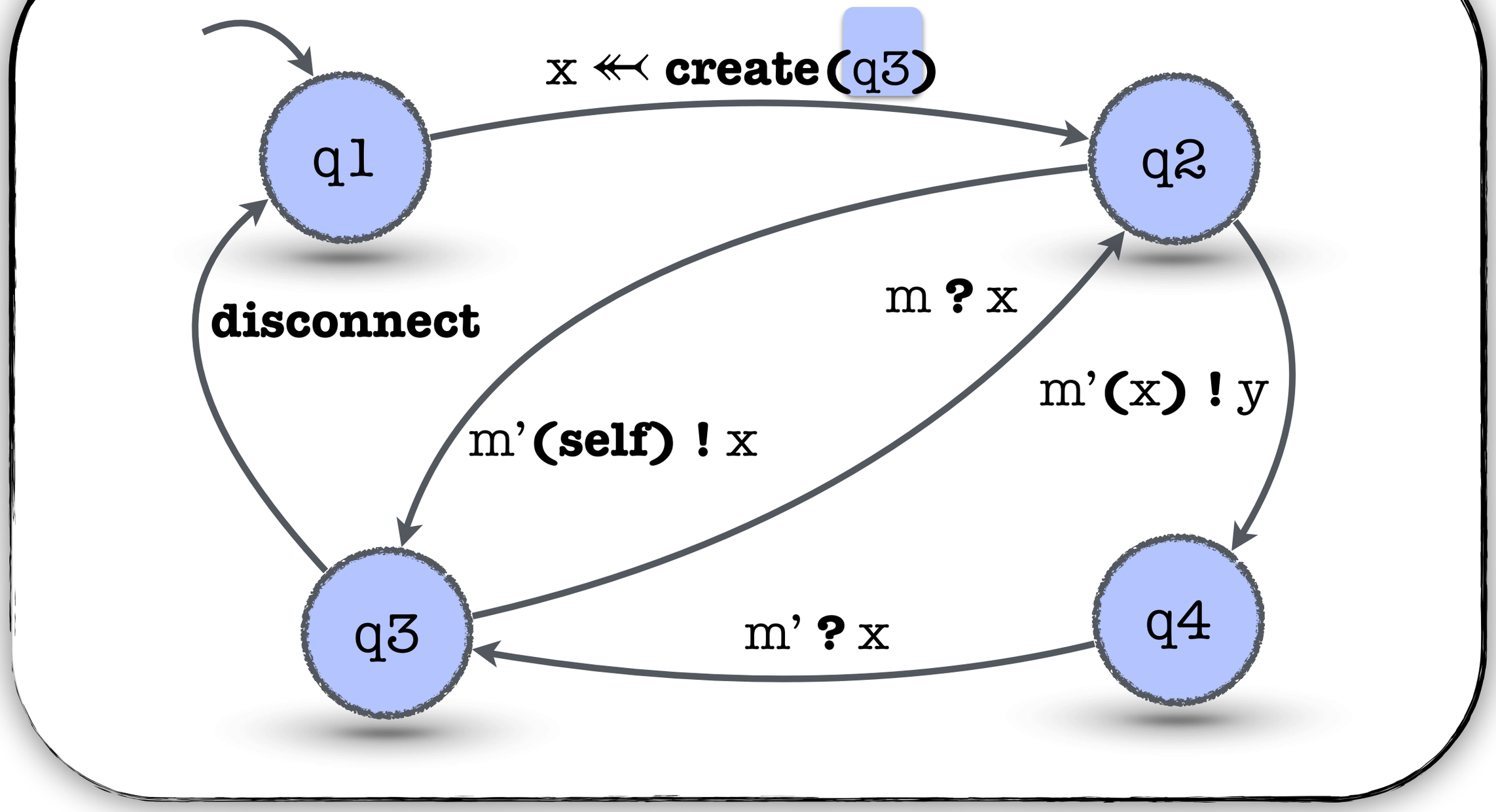
Process Model



Formal Model

Verification of Buffered Dynamic Register Automata

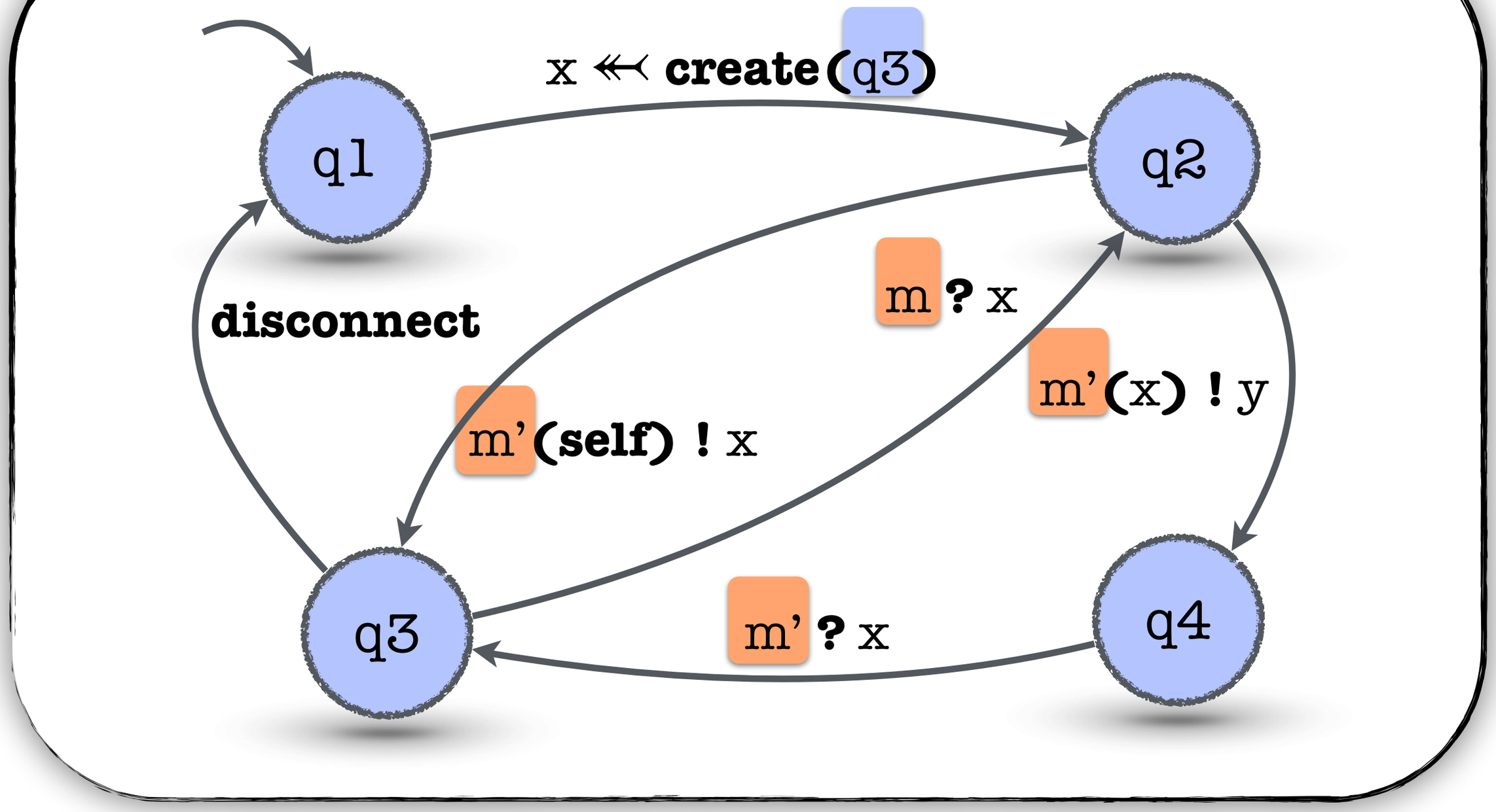
Process Model



Formal Model

Verification of Buffered Dynamic Register Automata

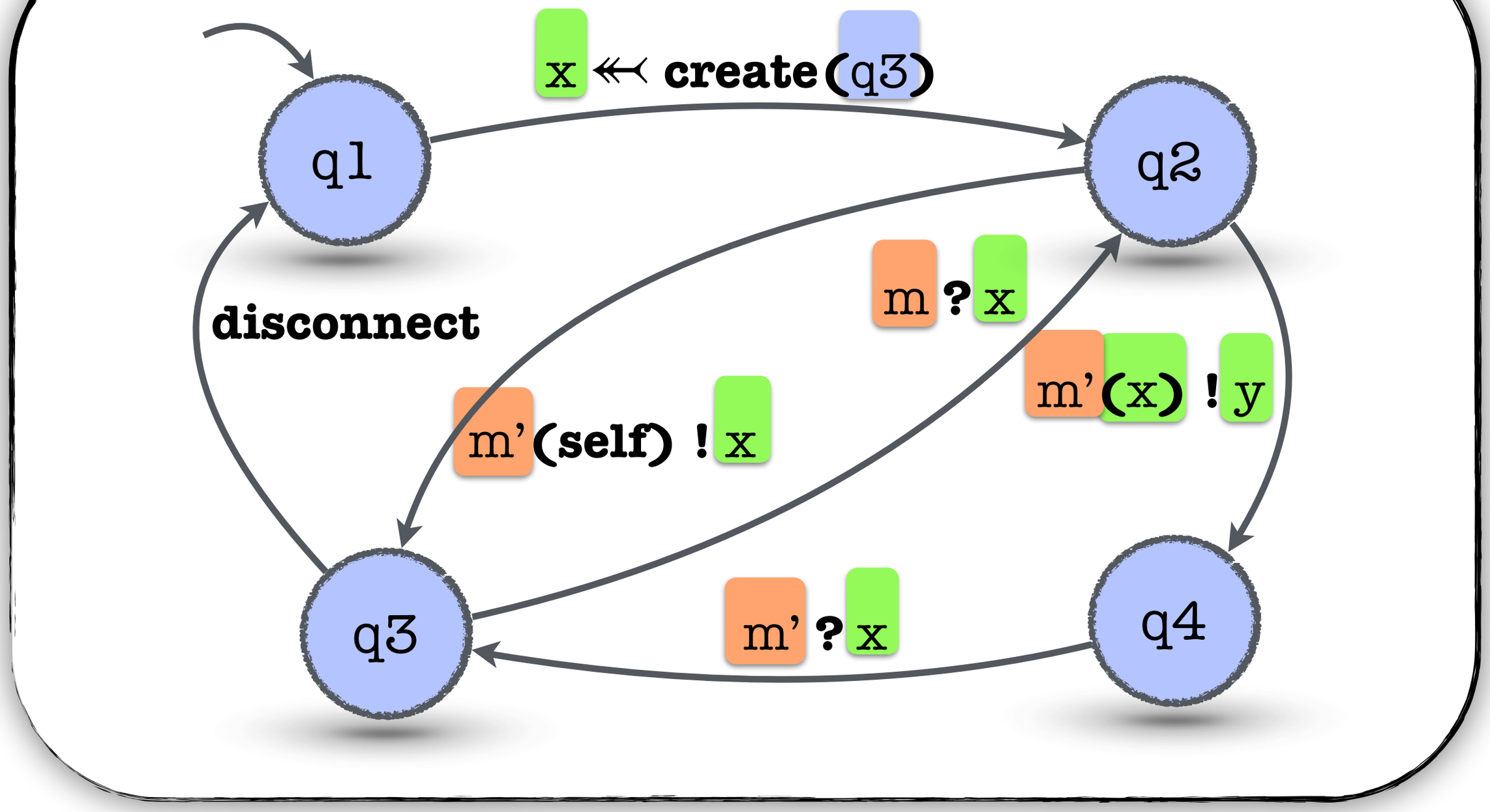
Process Model



Formal Model

Verification of Buffered Dynamic Register Automata

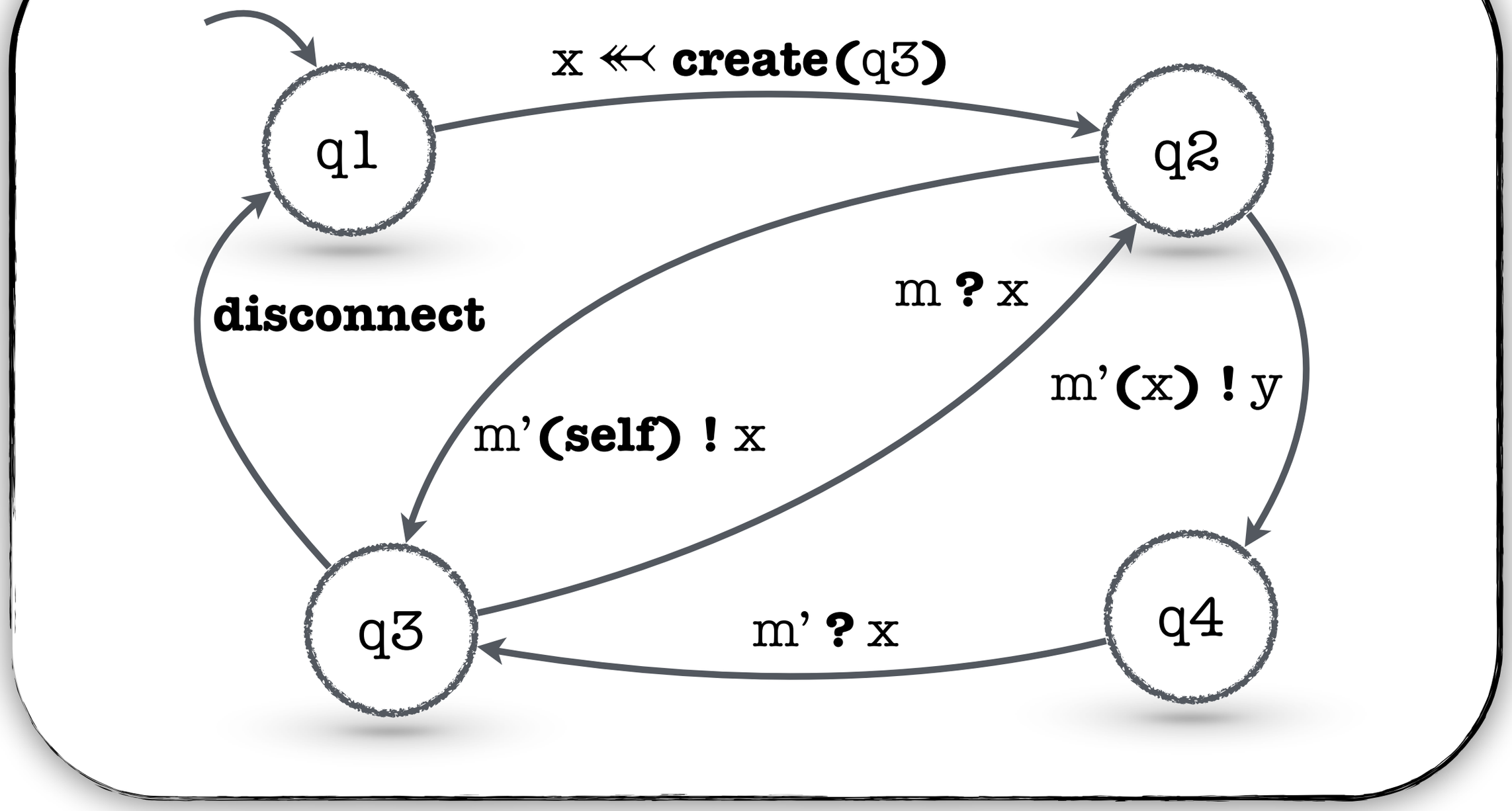
Process Model



Formal Model

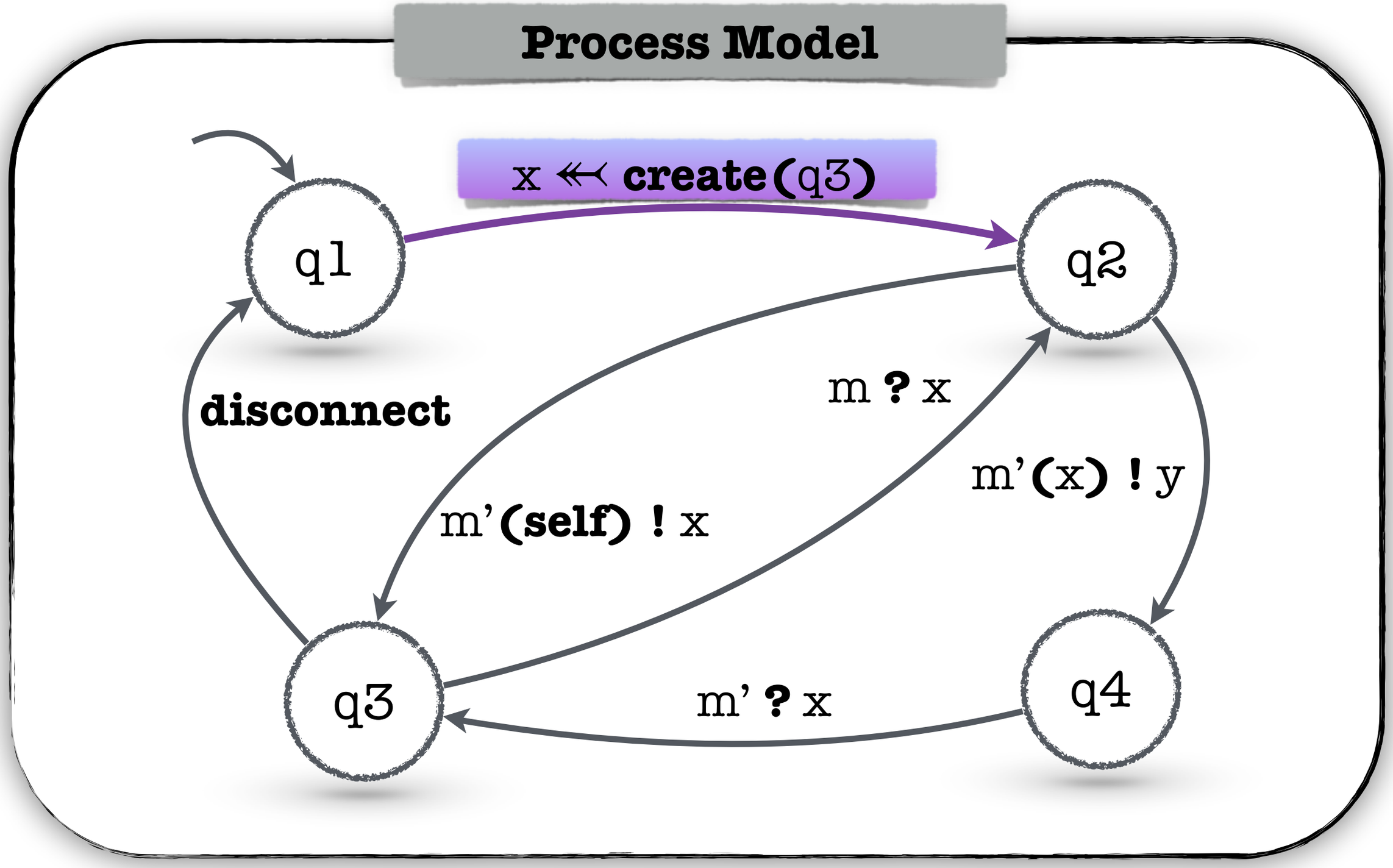
Verification of Buffered Dynamic Register Automata

Process Model



Formal Model

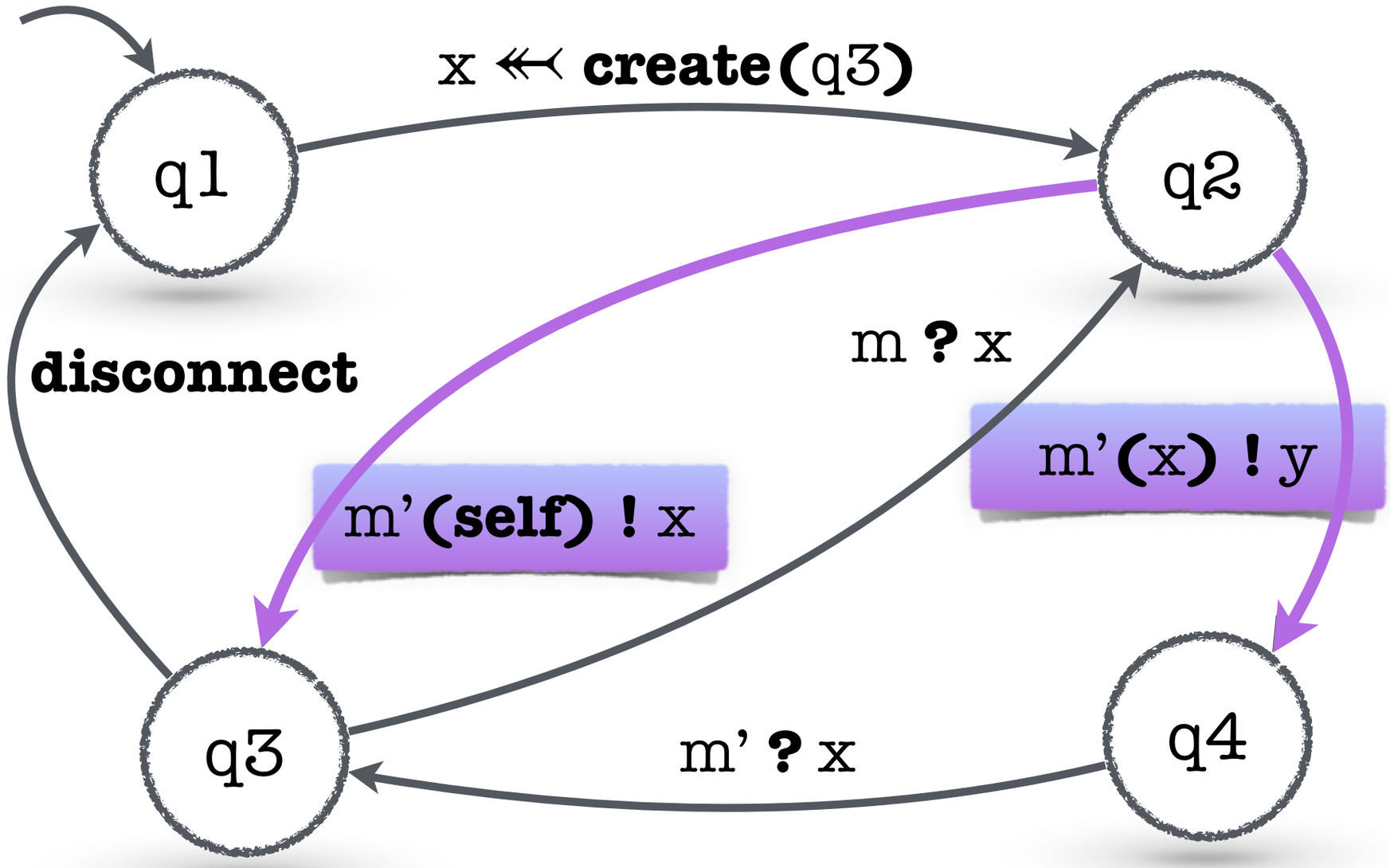
Verification of Buffered Dynamic Register Automata



Formal Model

Verification of Buffered Dynamic Register Automata

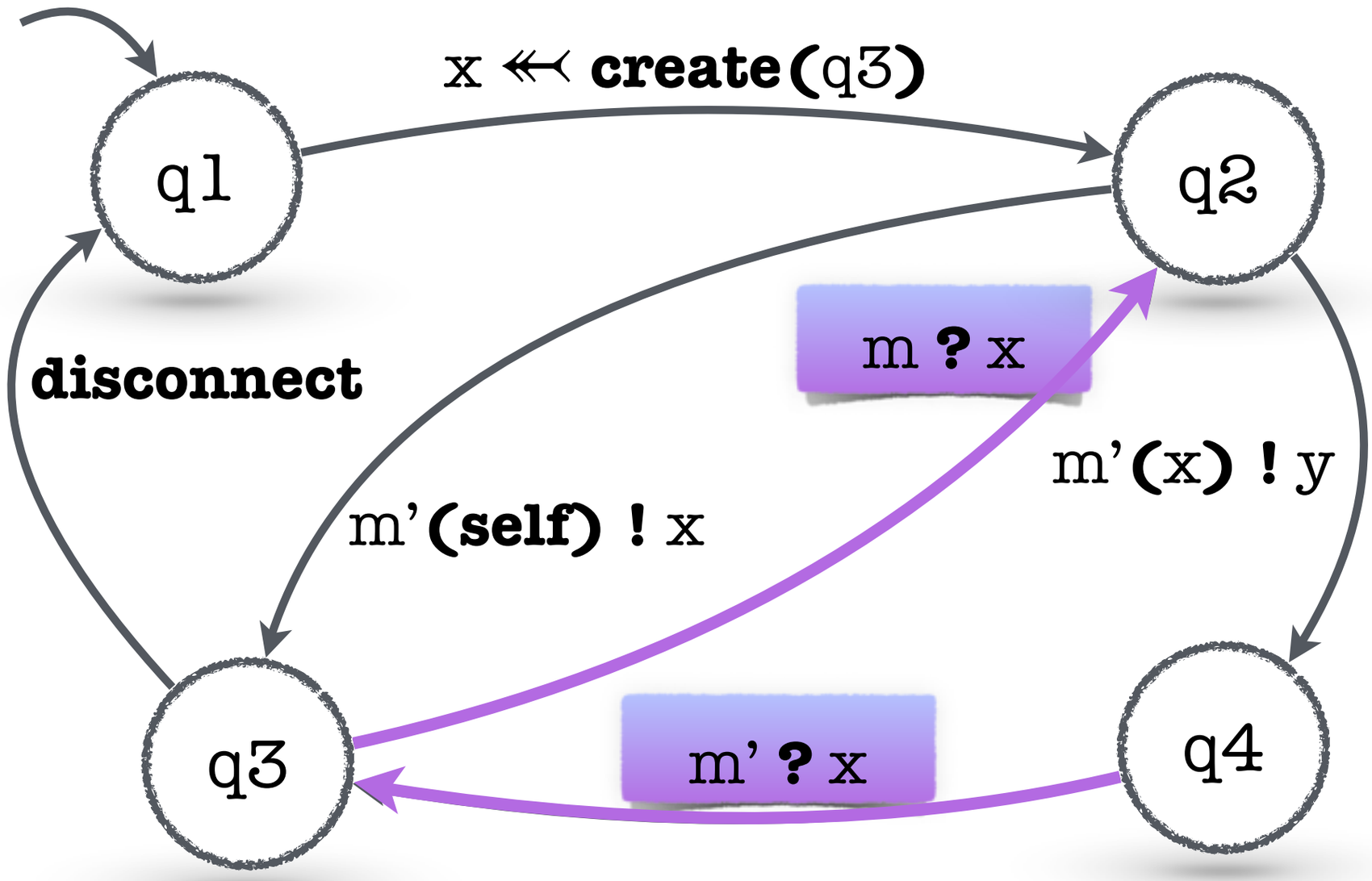
Process Model



Formal Model

Verification of Buffered Dynamic Register Automata

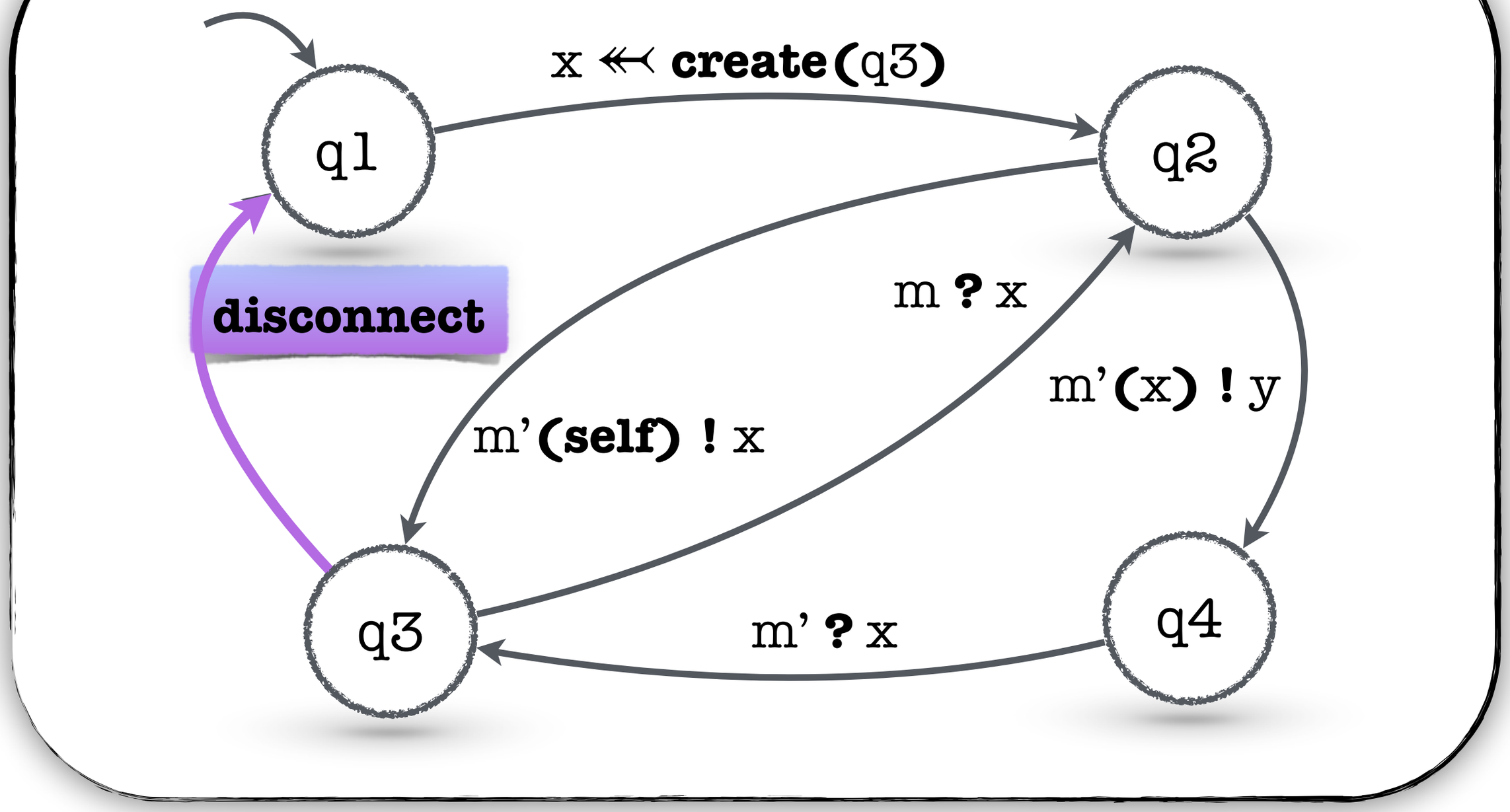
Process Model



Formal Model

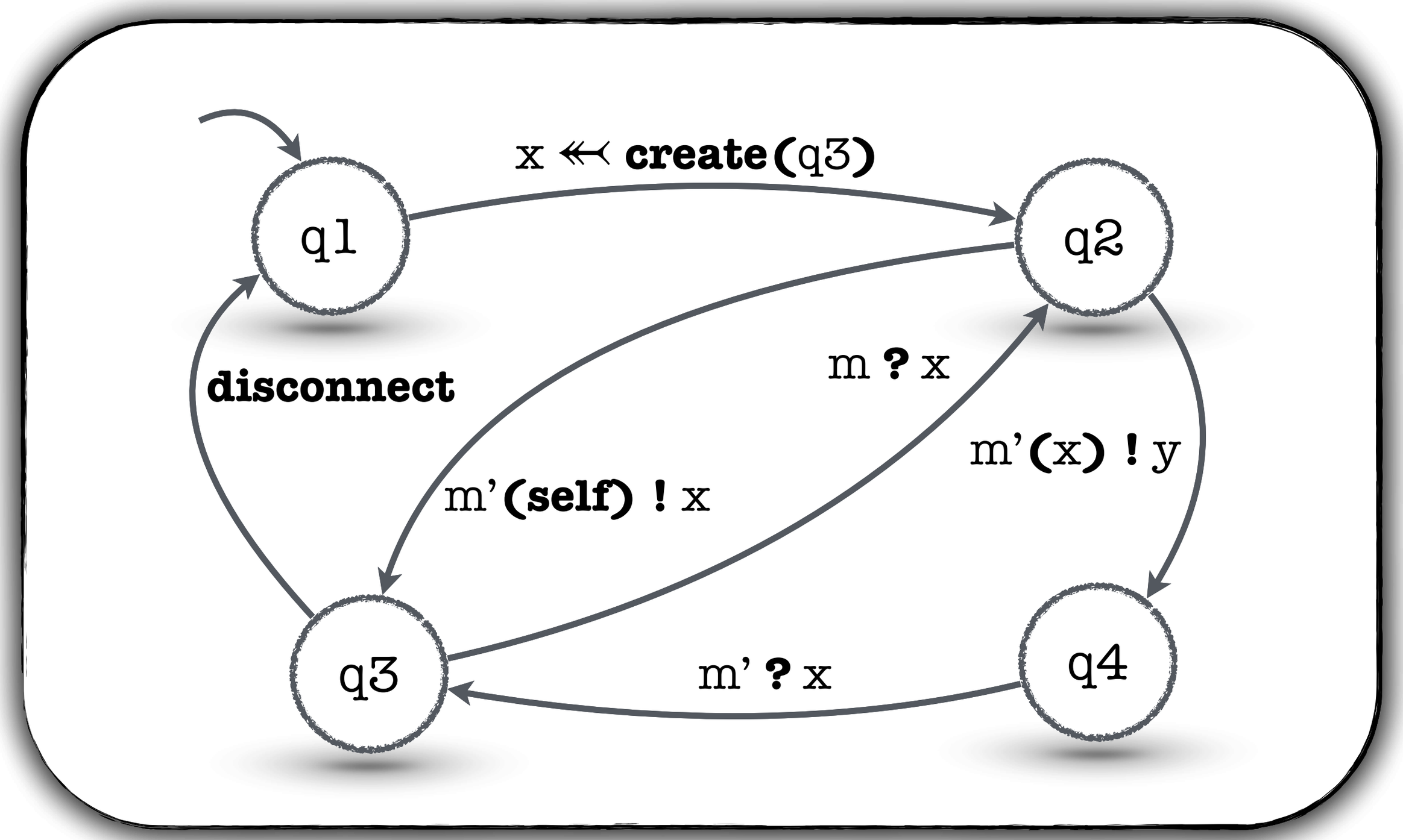
Verification of Buffered Dynamic Register Automata

Process Model



Formal Model

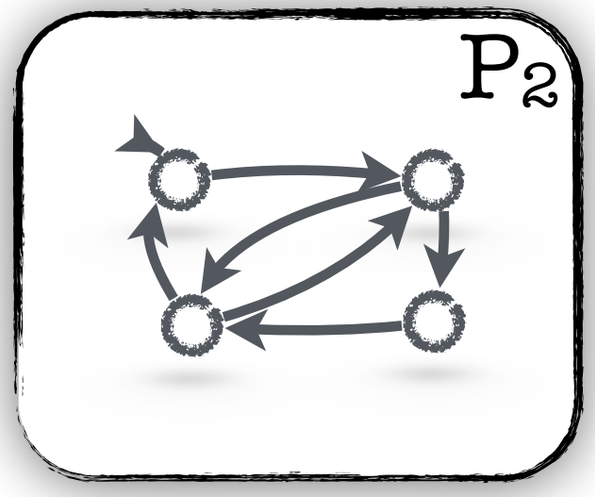
Process Model

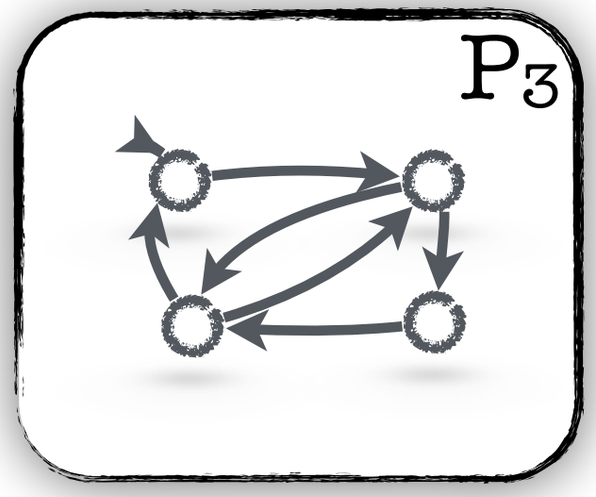


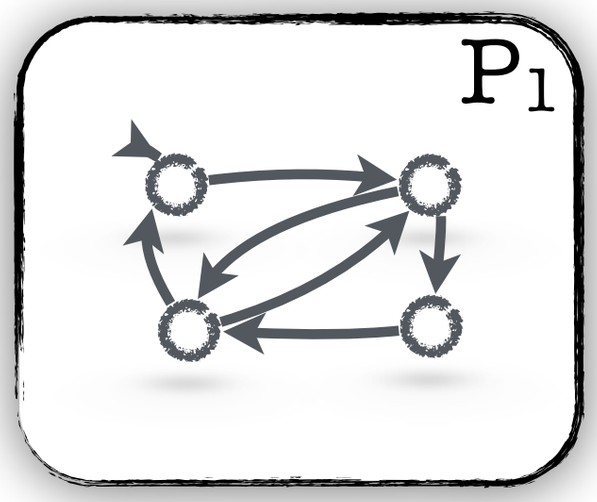
Formal Model

Verification of Buffered Dynamic Register Automata

Process Model **Configuration**



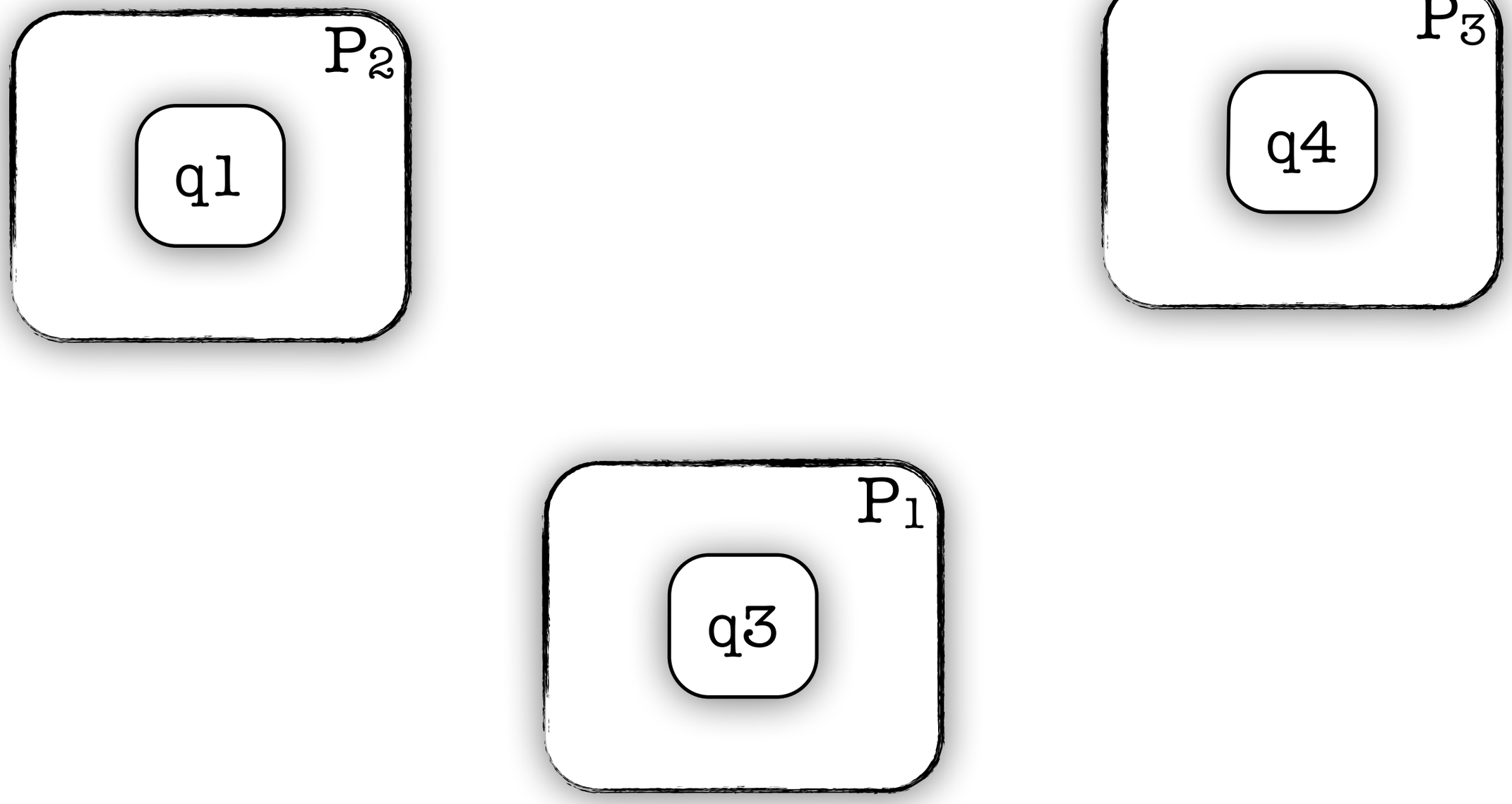




Formal Model

Verification of Buffered Dynamic Register Automata

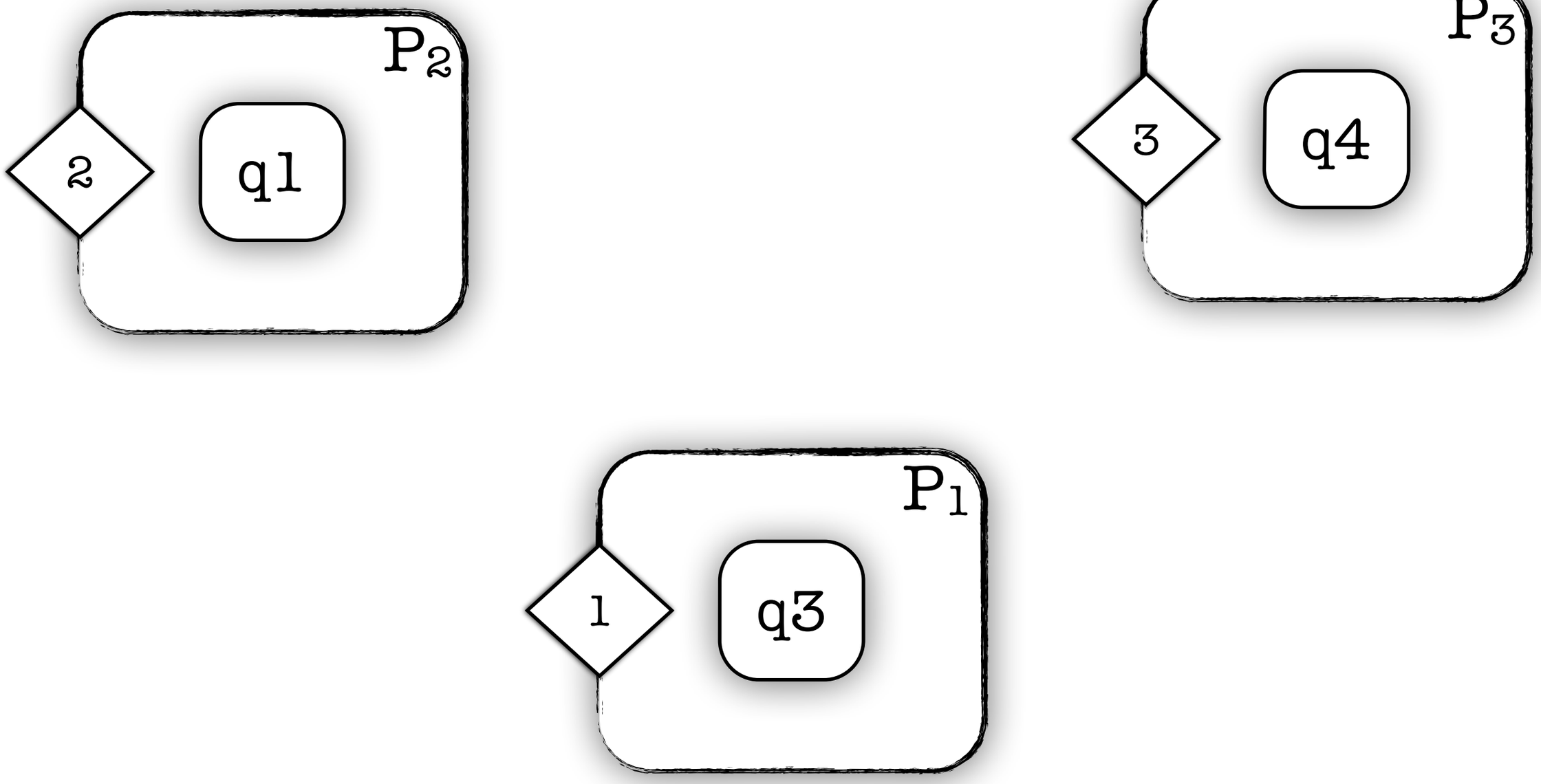
Process Model **Configuration**



Formal Model

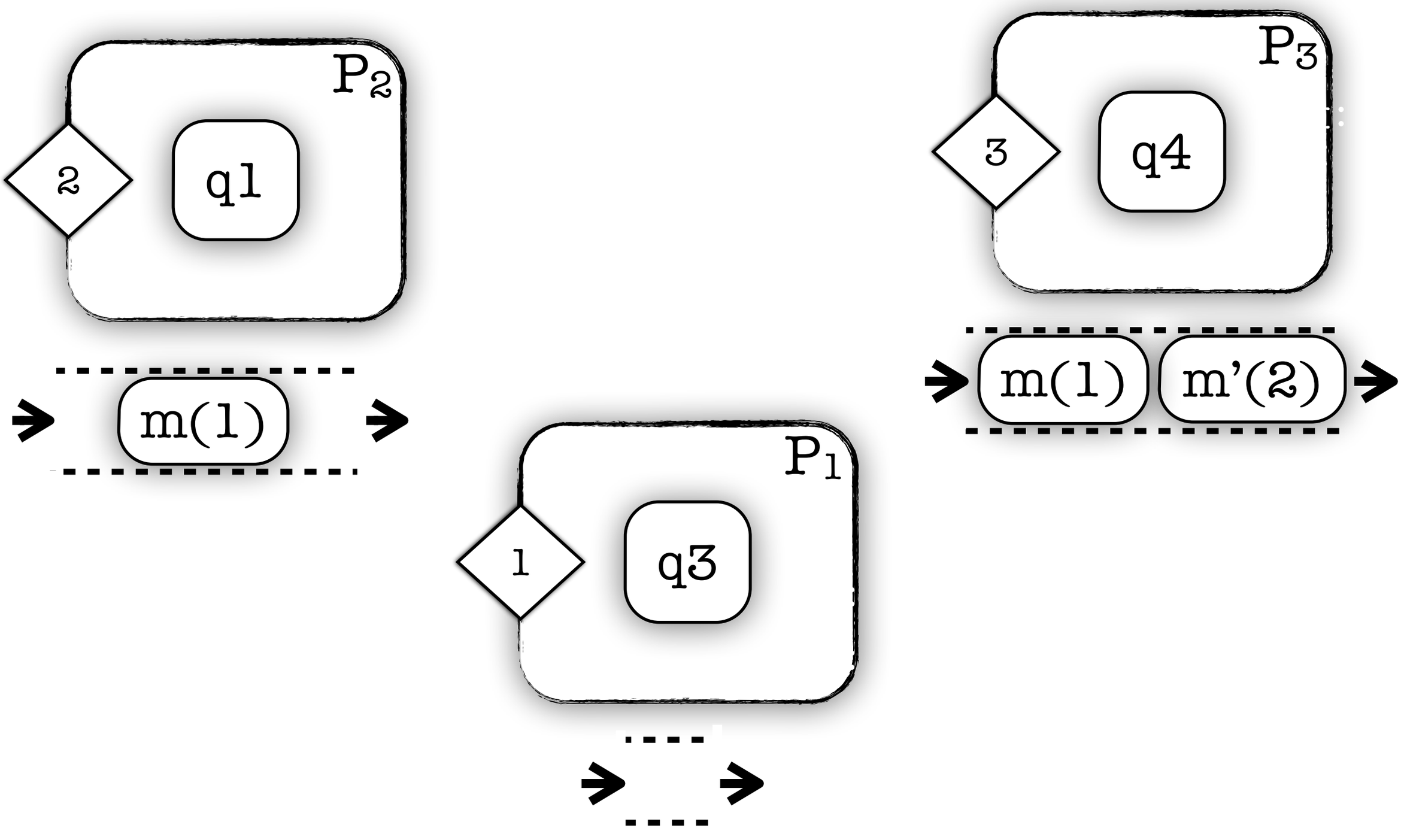
Verification of Buffered Dynamic Register Automata

Process Model **Configuration**



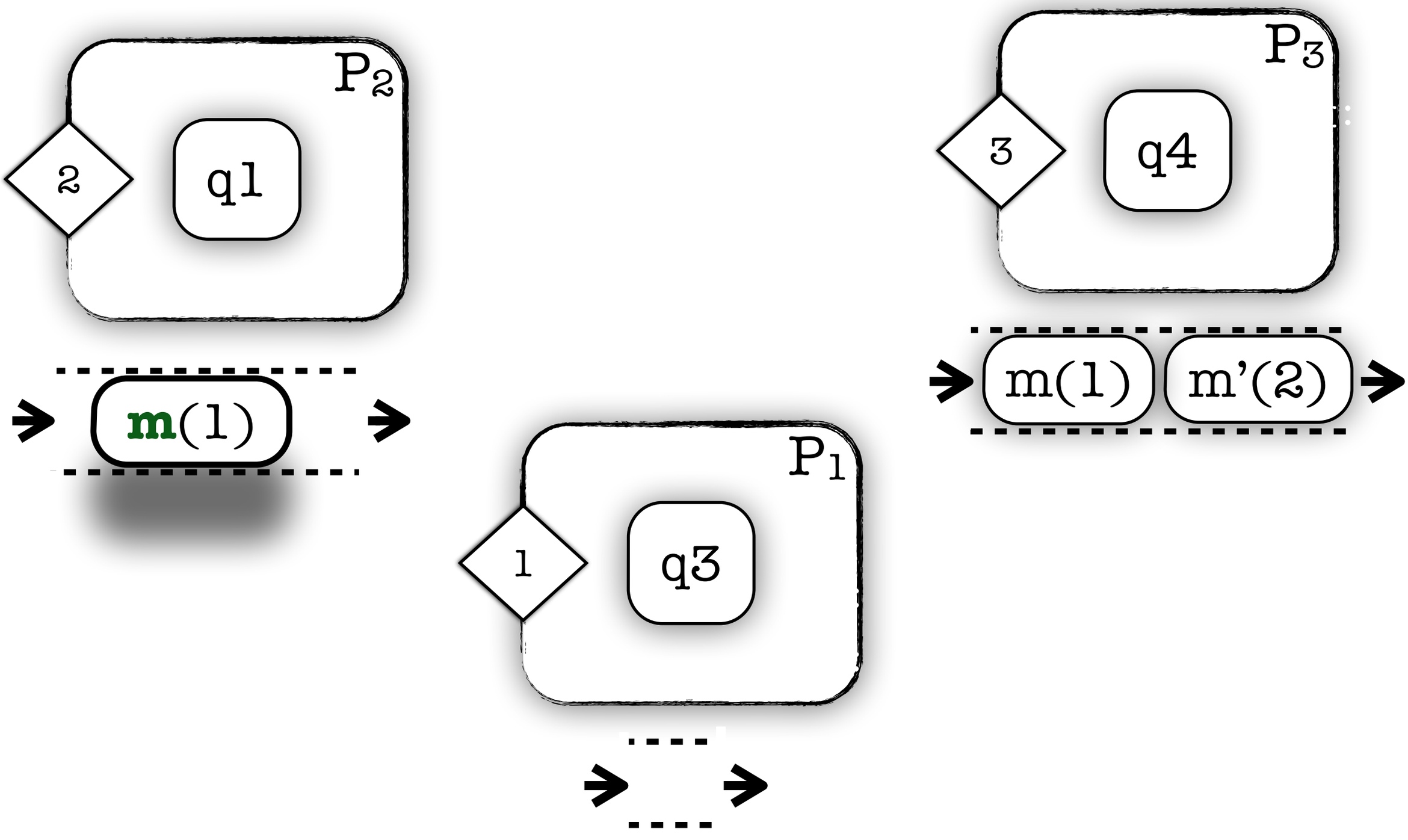
Formal Model

Process Model **Configuration**



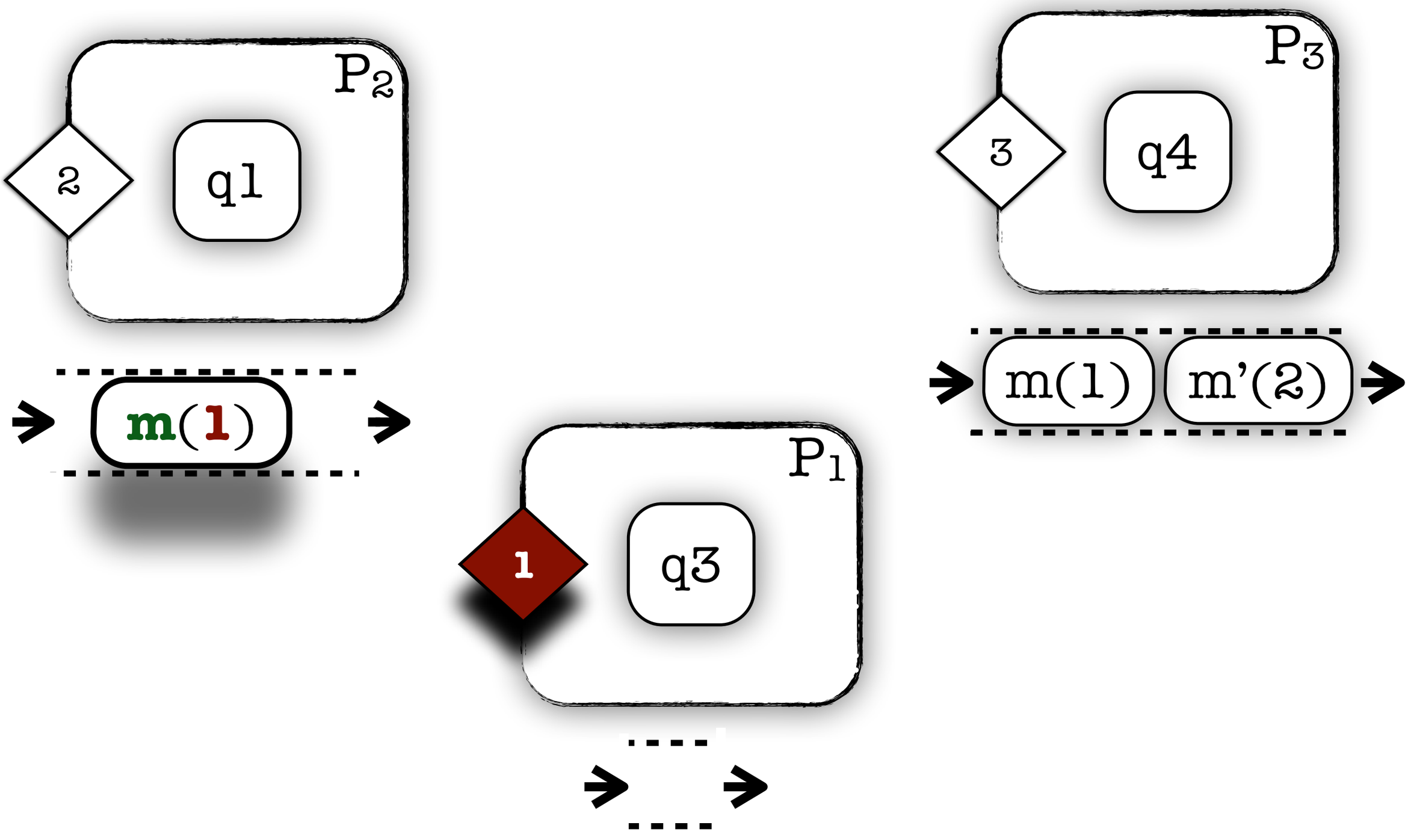
Formal Model

Process Model **Configuration**



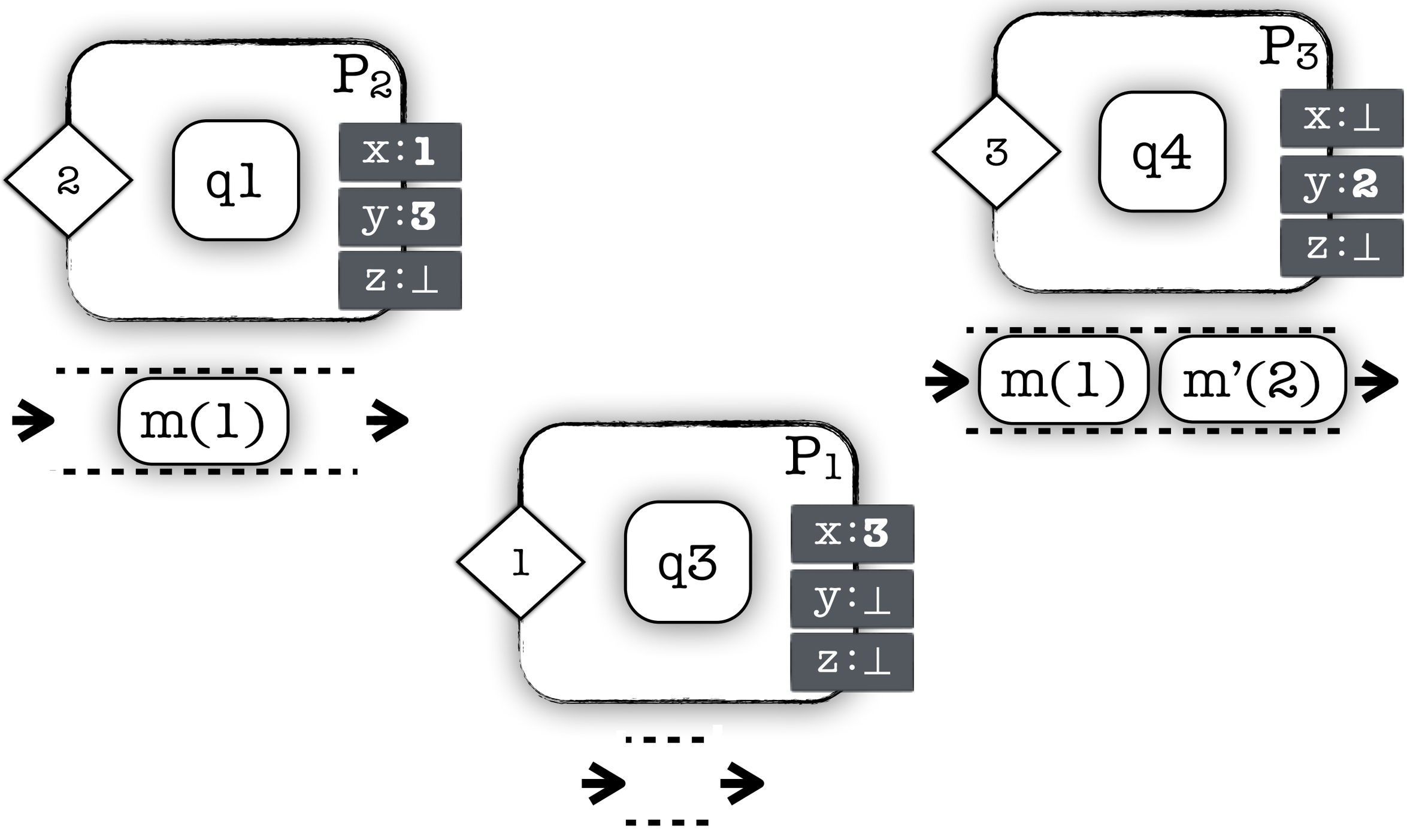
Formal Model

Process Model **Configuration**



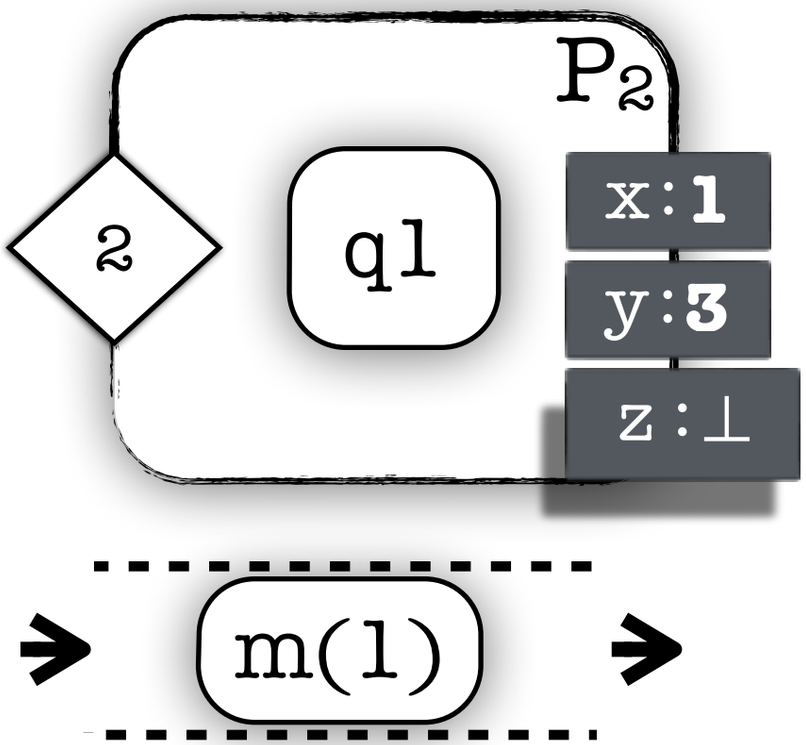
Formal Model

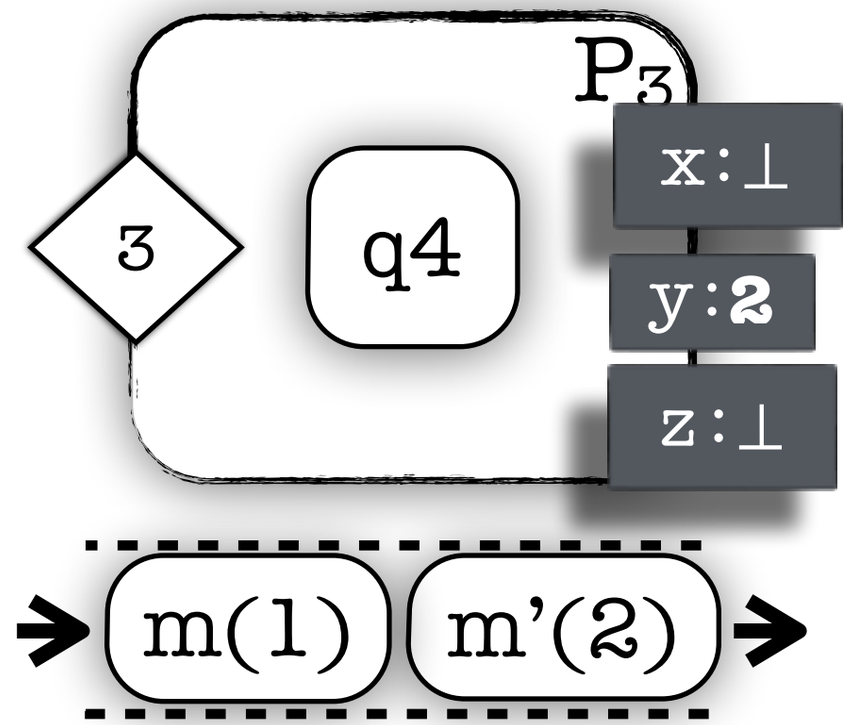
Process Model **Configuration**

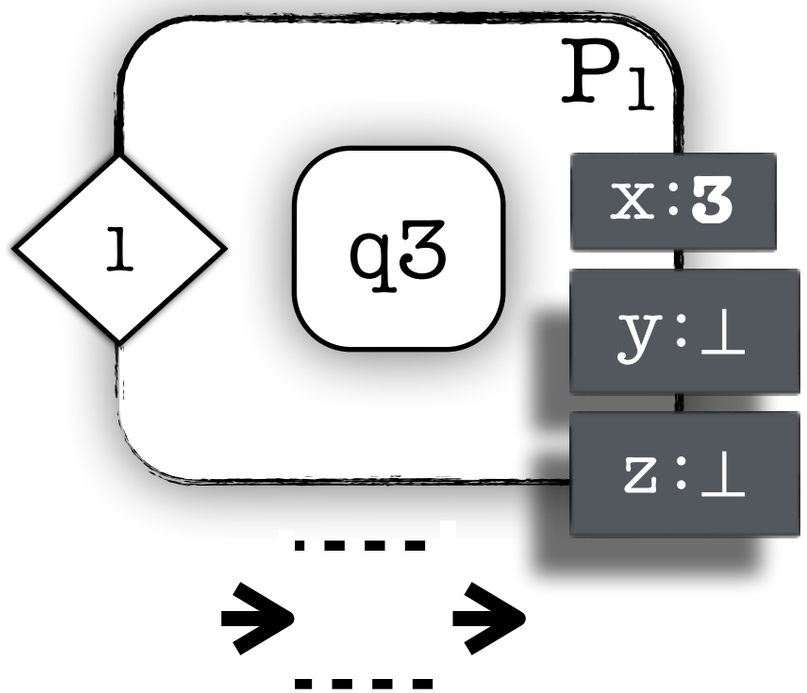


Formal Model

Process Model **Configuration**

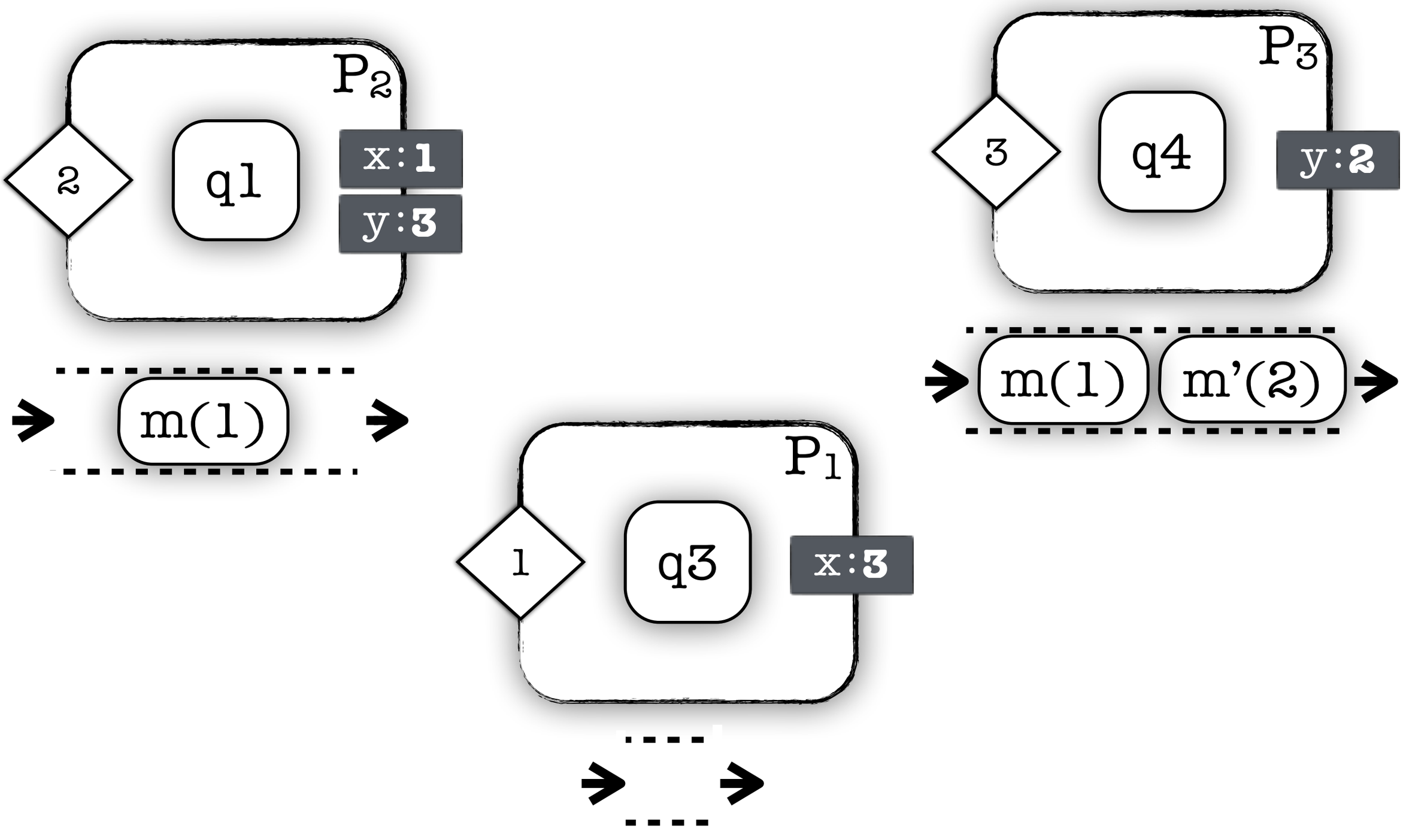






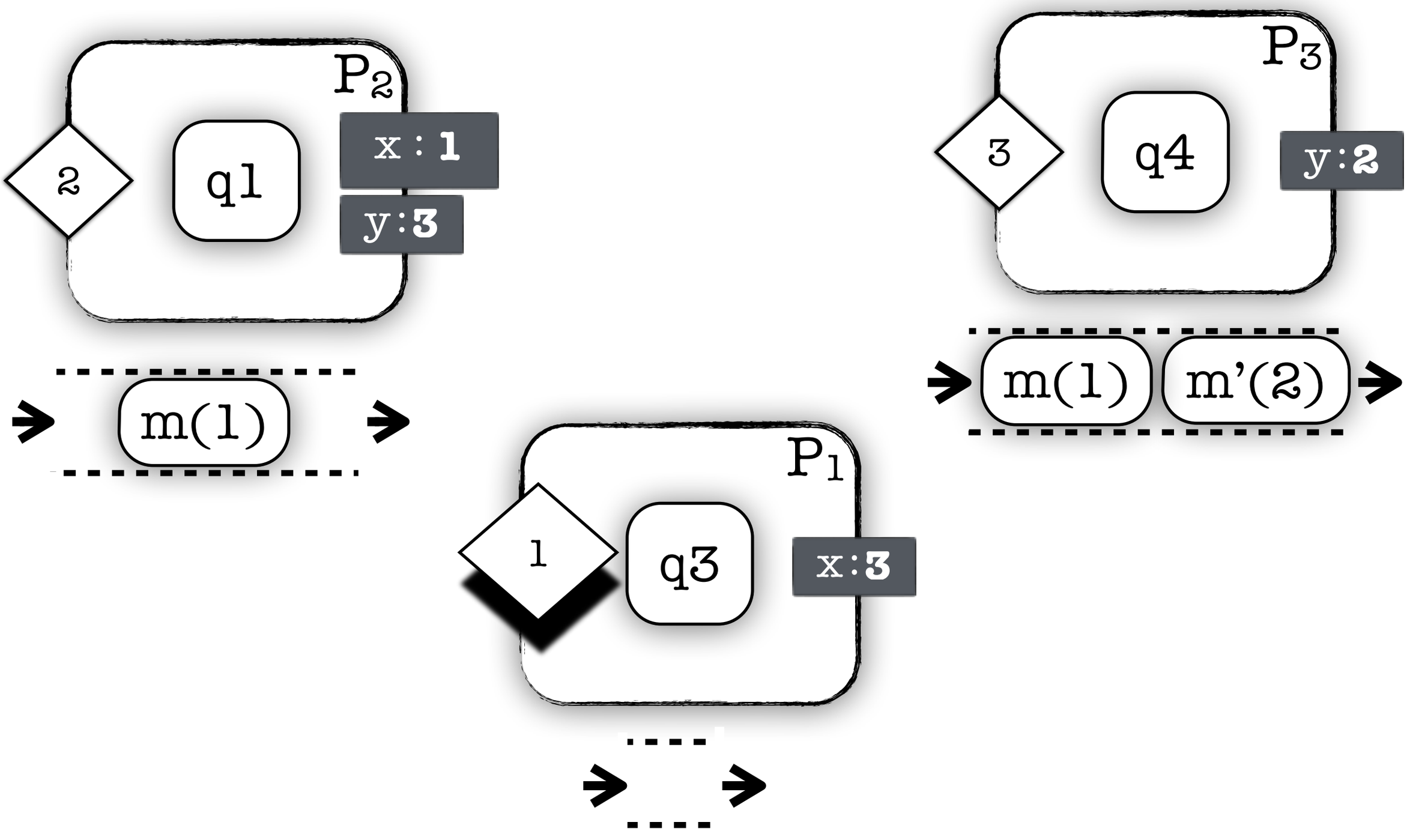
Formal Model

Process Model **Configuration**



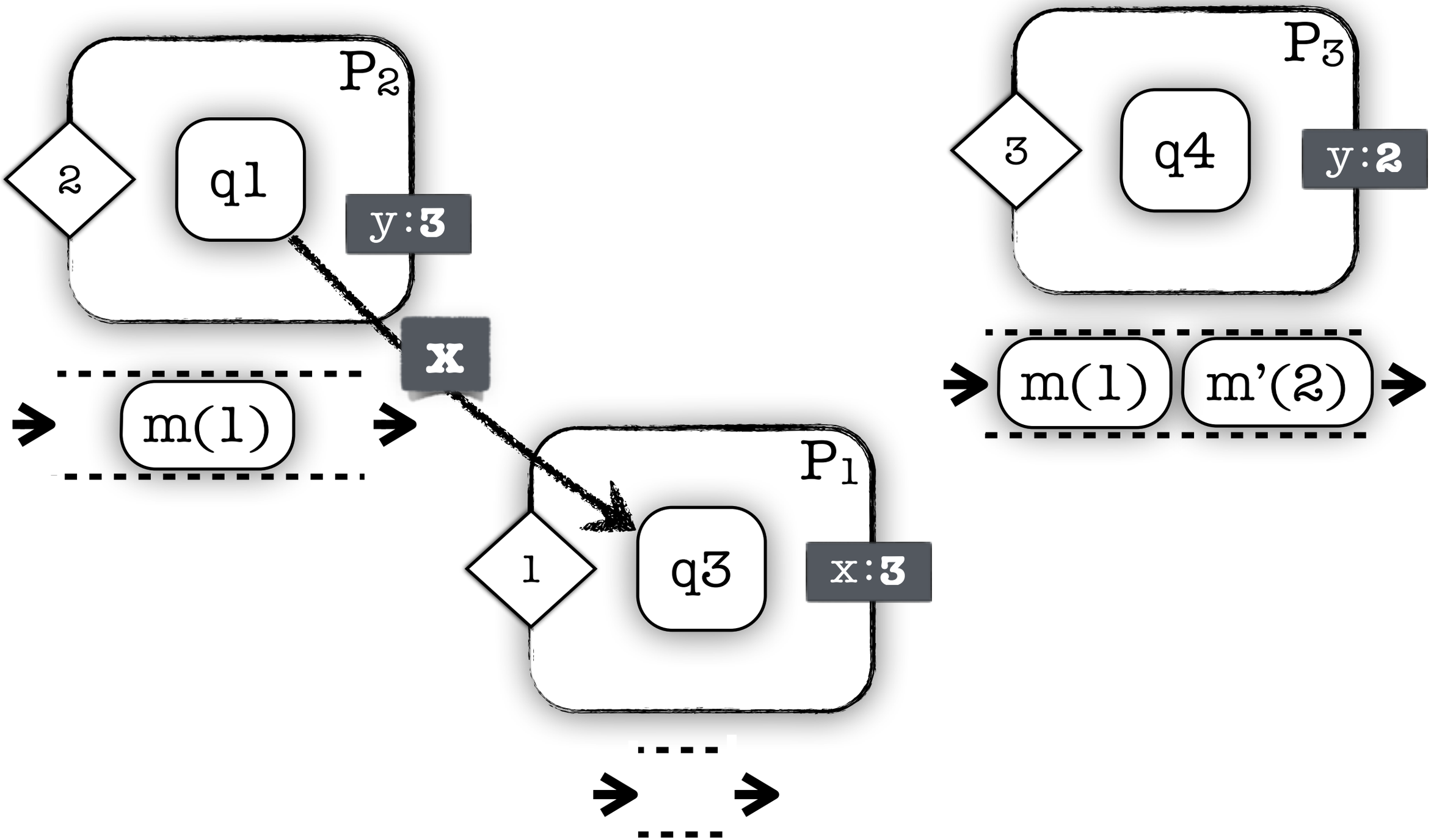
Formal Model

Process Model **Configuration**



Formal Model

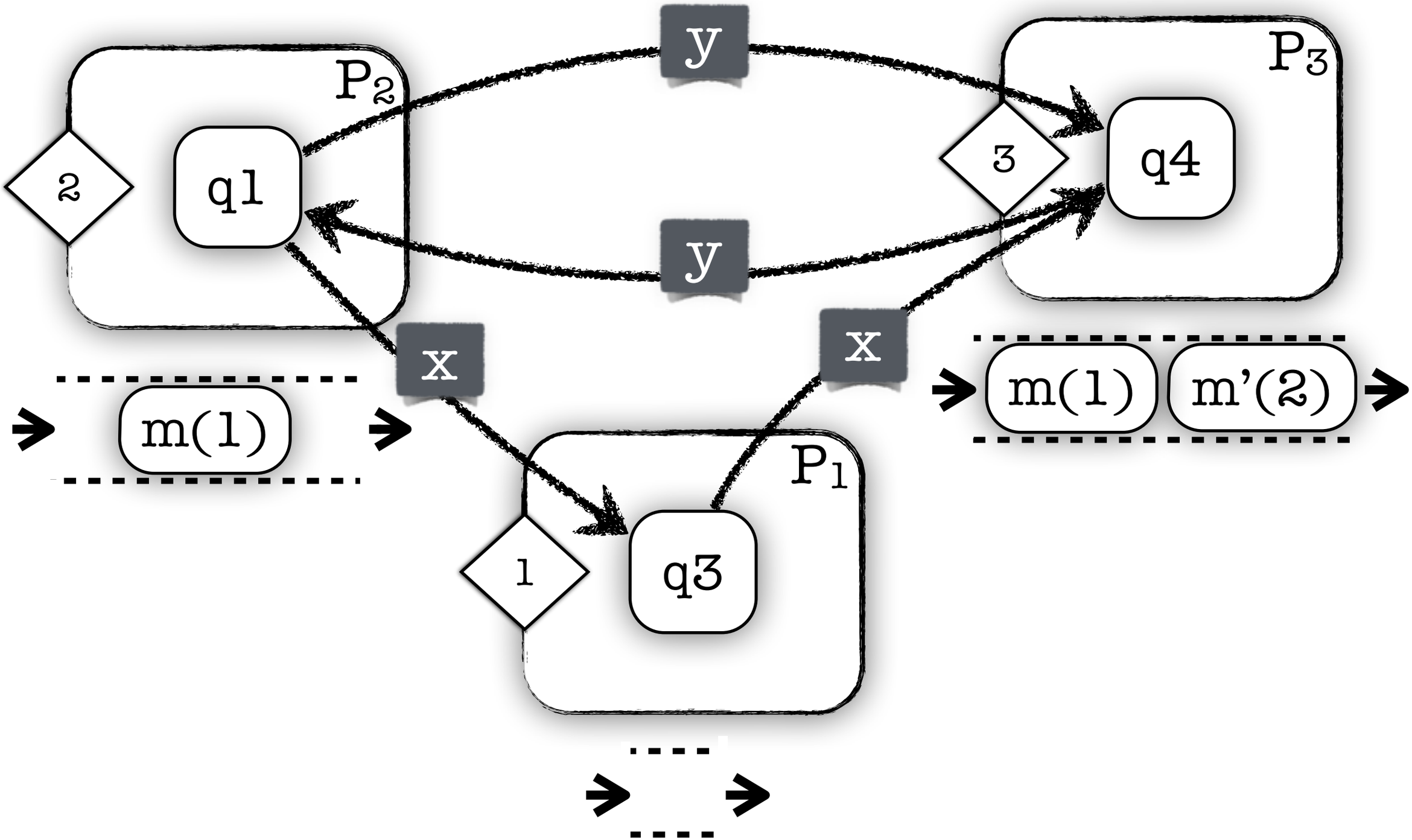
Process Model **Configuration**



Formal Model

Verification of Buffered Dynamic Register Automata

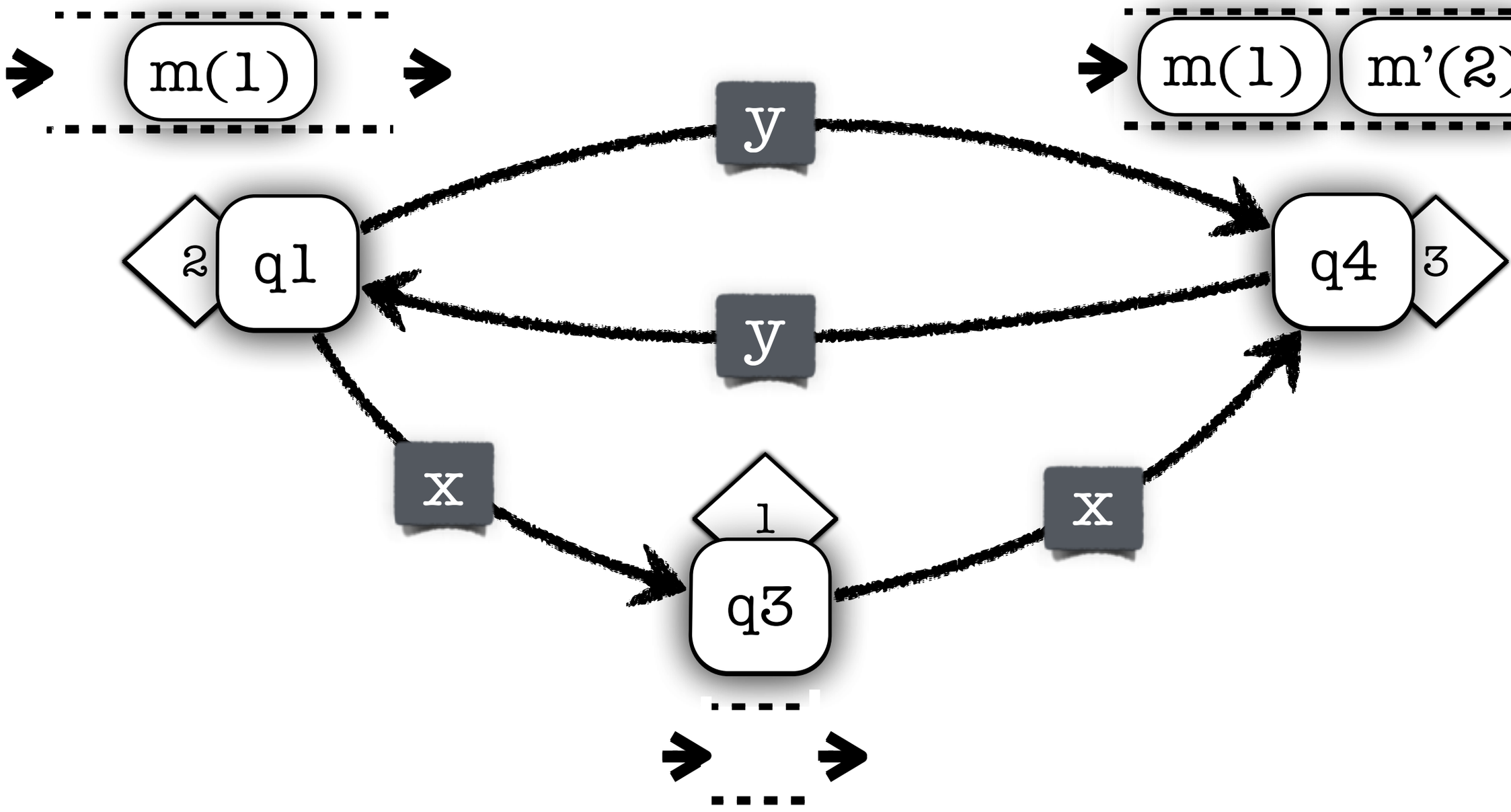
Process Model **Configuration**



Formal Model

Verification of Buffered Dynamic Register Automata

Process Model | **Configuration** | **Configuration Graph**

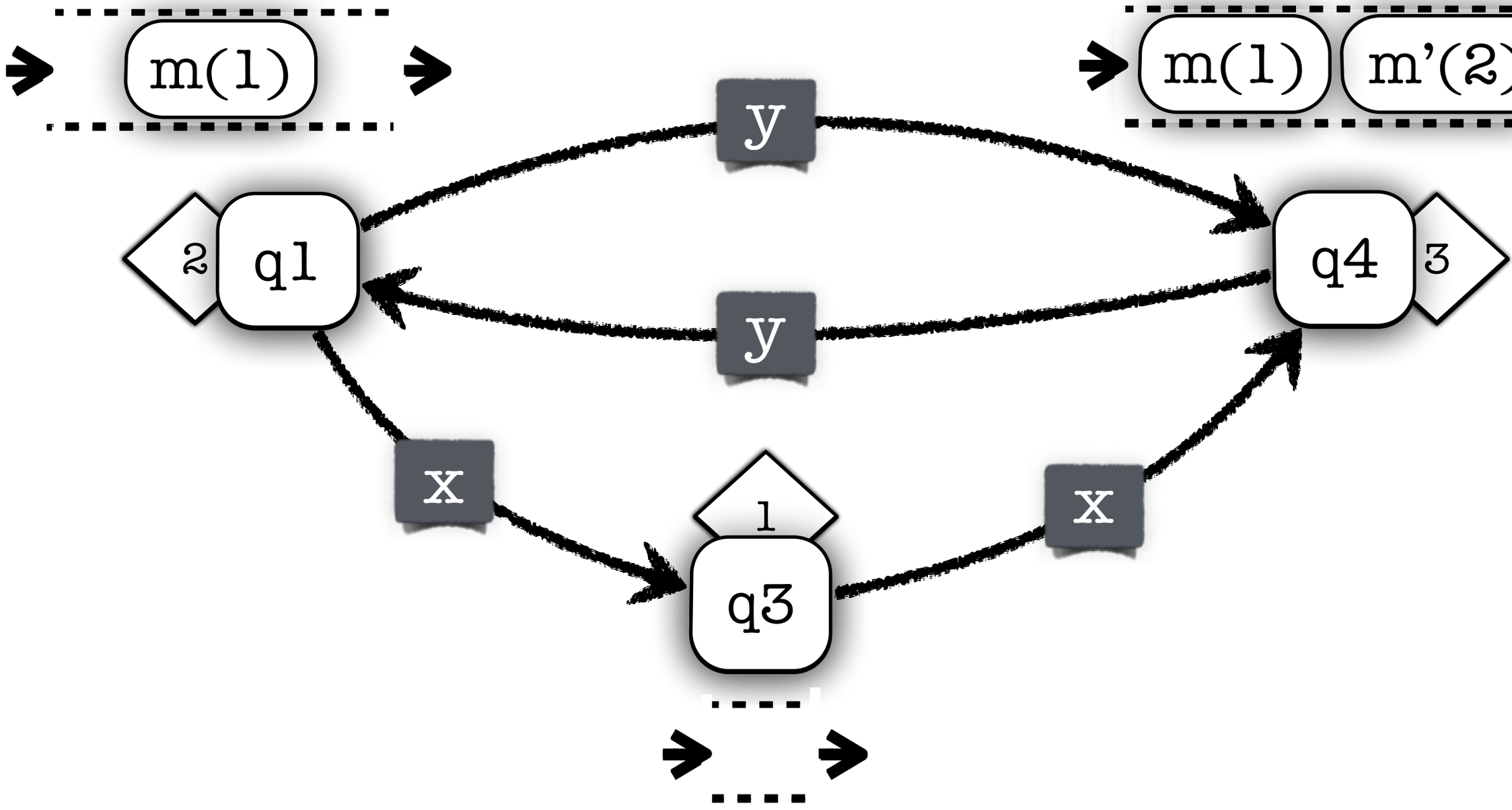


Formal Model

Verification of Buffered Dynamic Register Automata

Process Model **Configuration** **Configuration Graph**

Communication Graph



Buffered Dynamic Register Automata

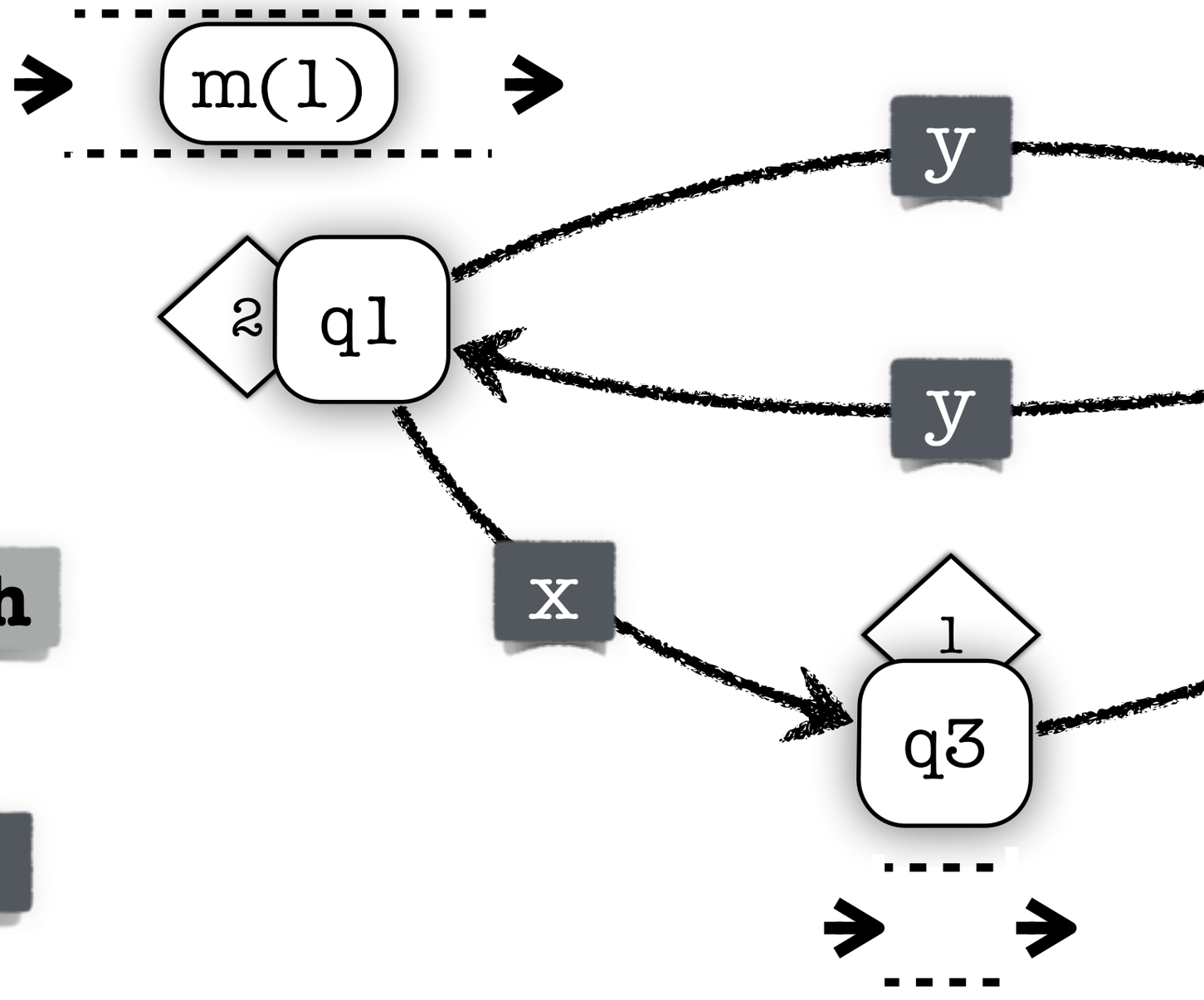
Formal Model

Process Model

Configuration

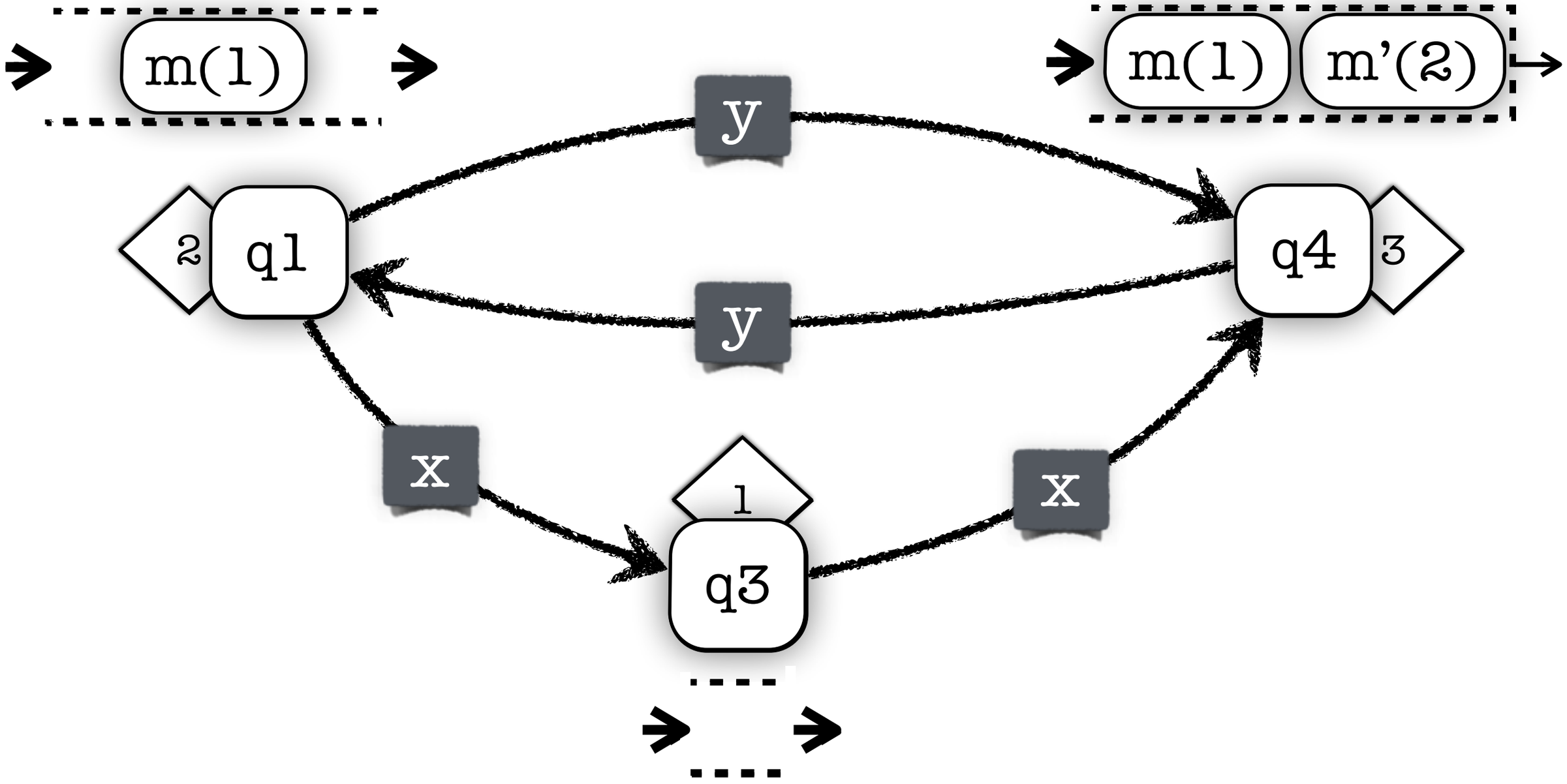
Configuration Graph

Operational Semantics



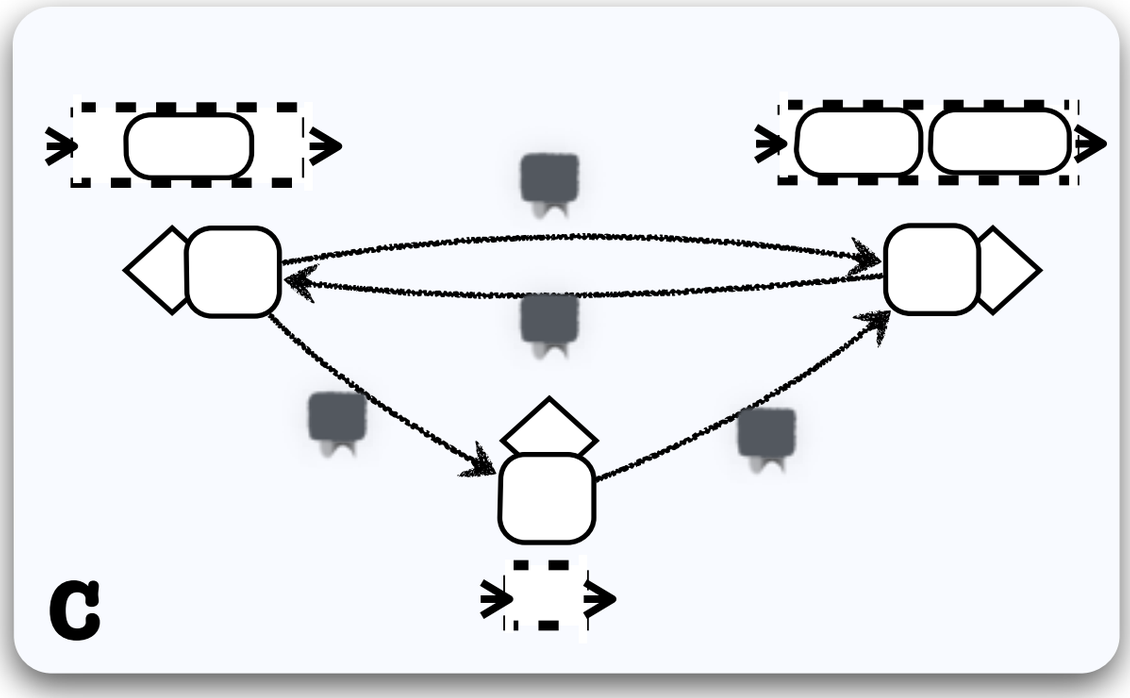
Operational Semantics

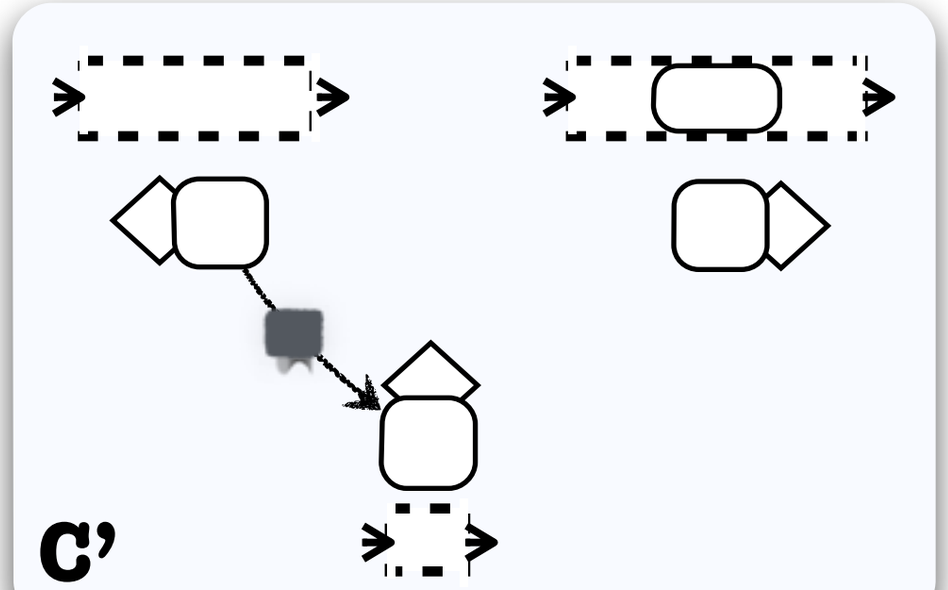
Verification of Buffered Dynamic Register Automata



Operational Semantics

Verification of Buffered Dynamic Register Automata





Operational Semantics

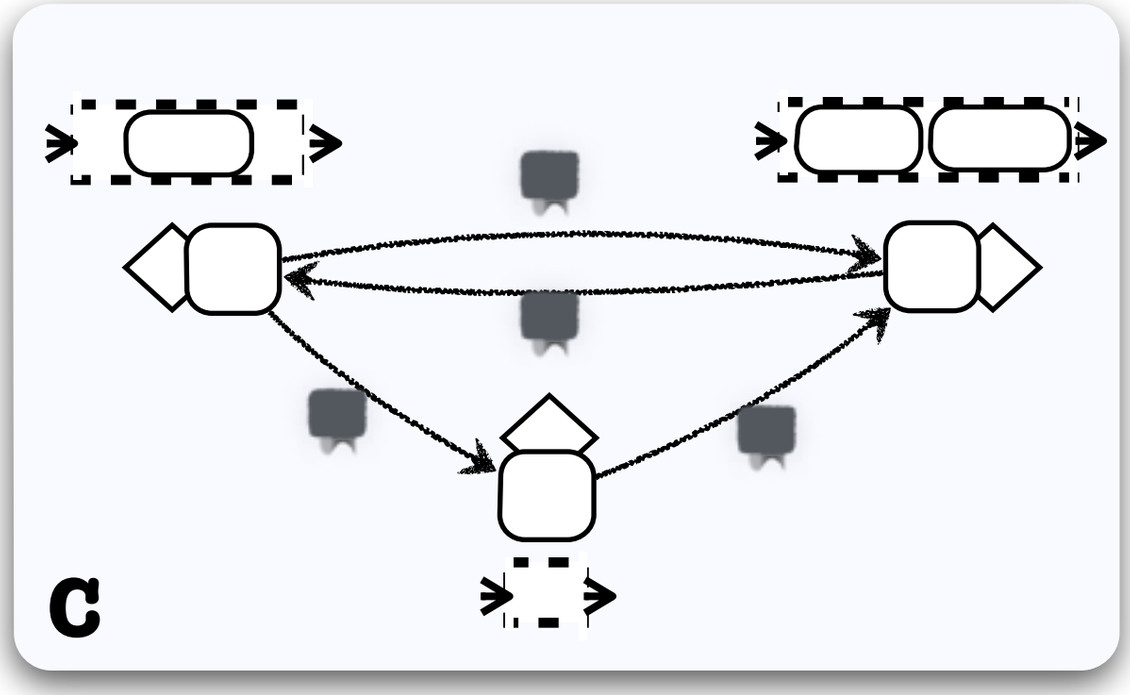
Verification of Buffered Dynamic Register Automata

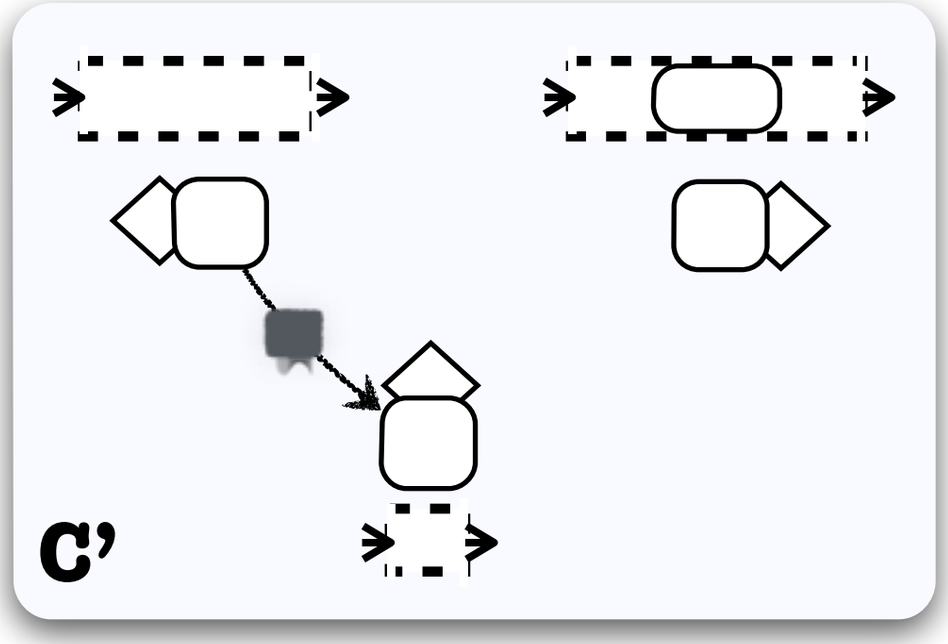
Process Creation

Process Disconnection

Message Sending

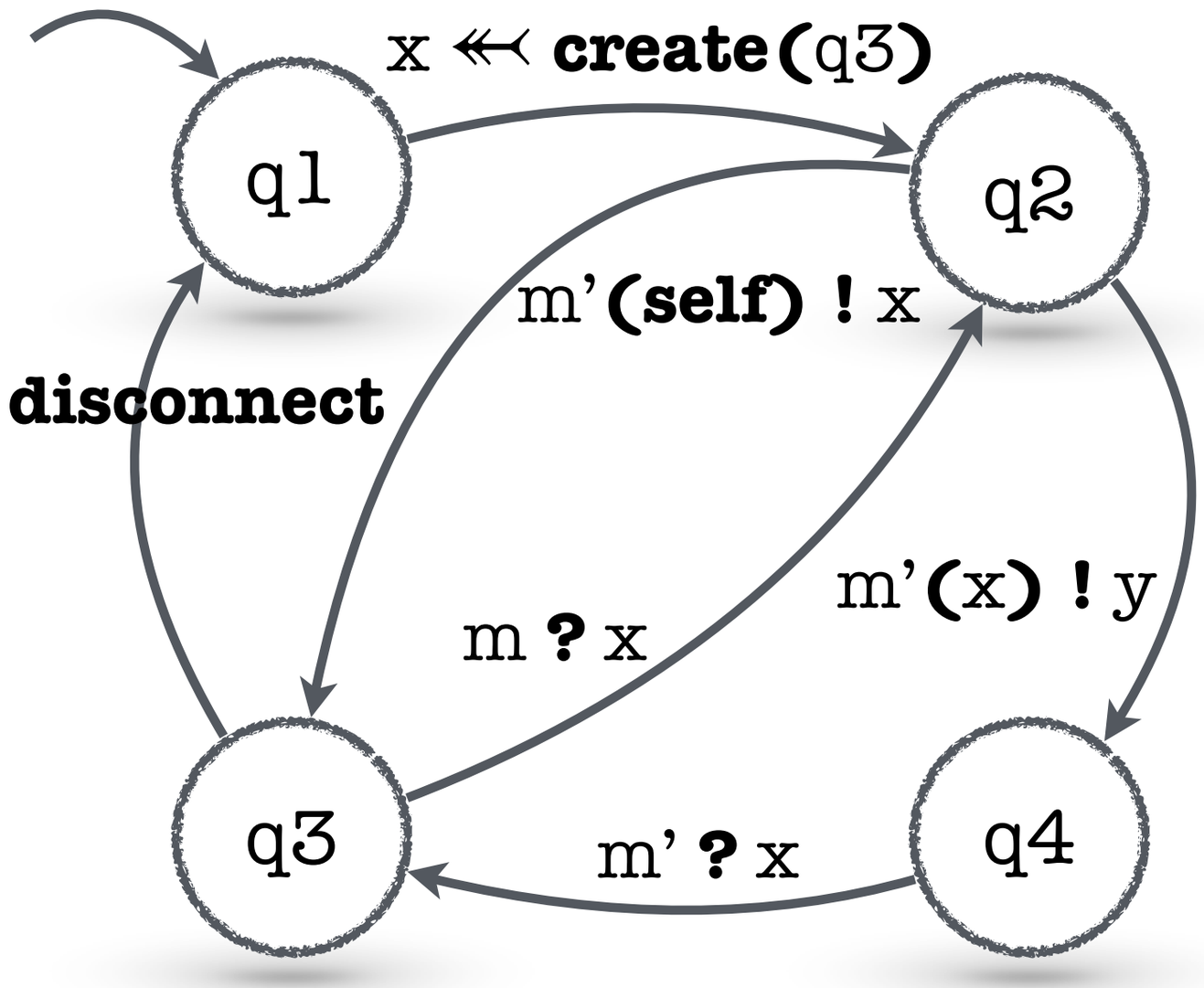
Message Receiving





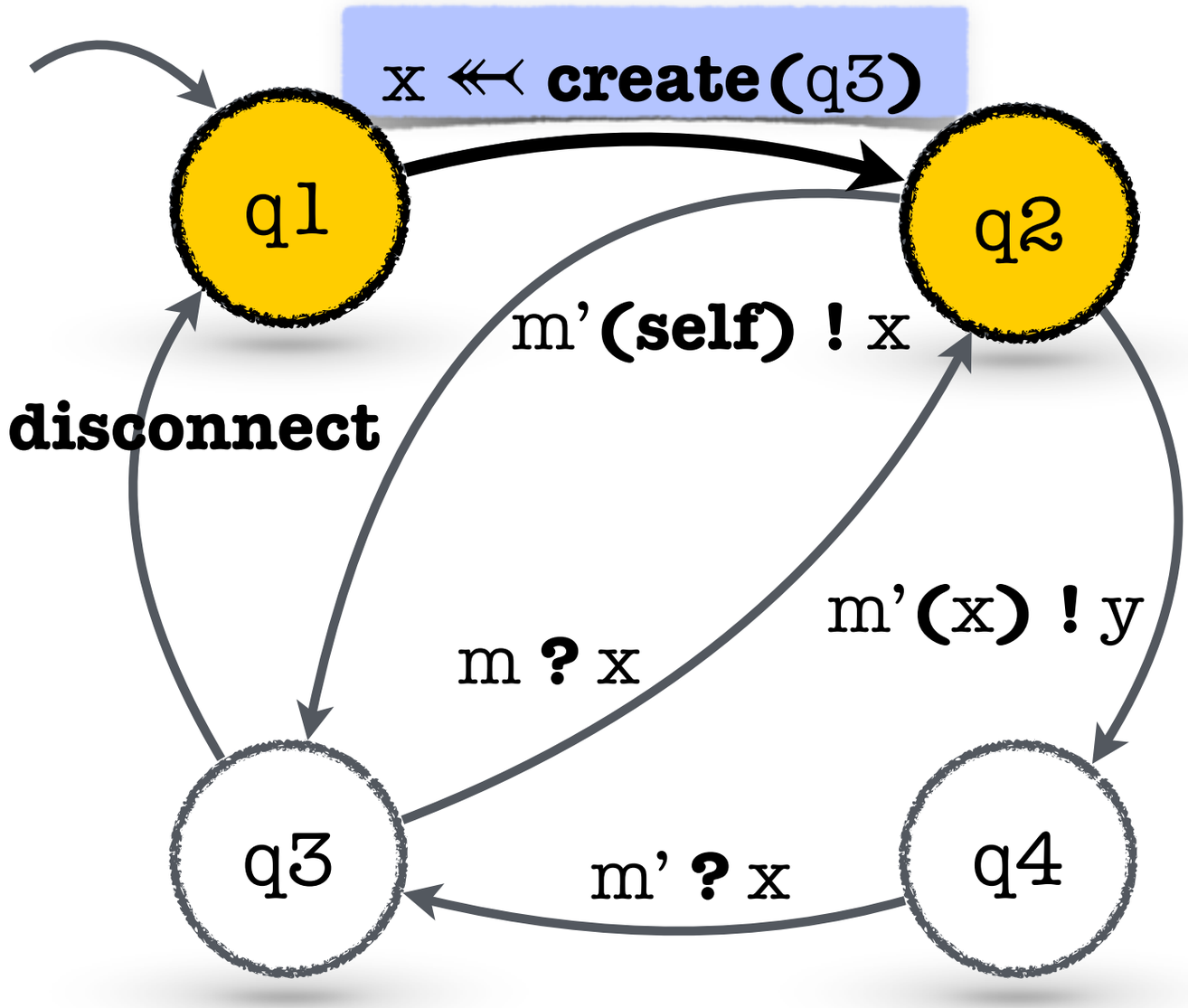
Operational Semantics

Process Creation



Operational Semantics

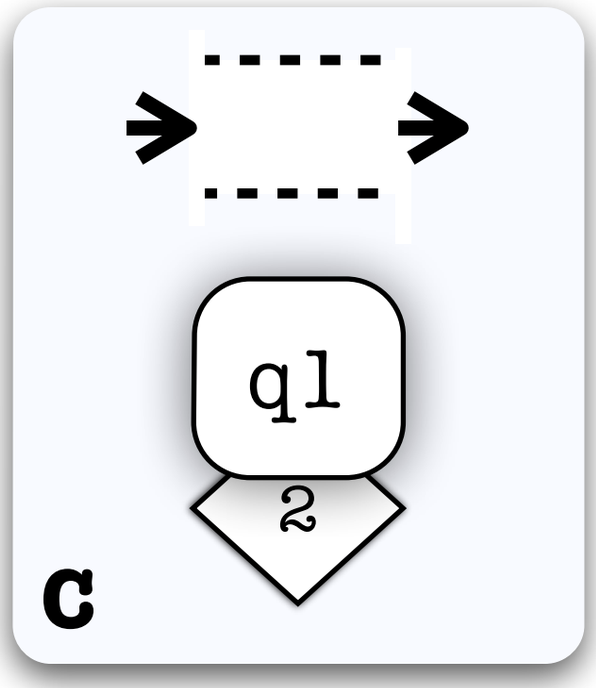
Process Creation

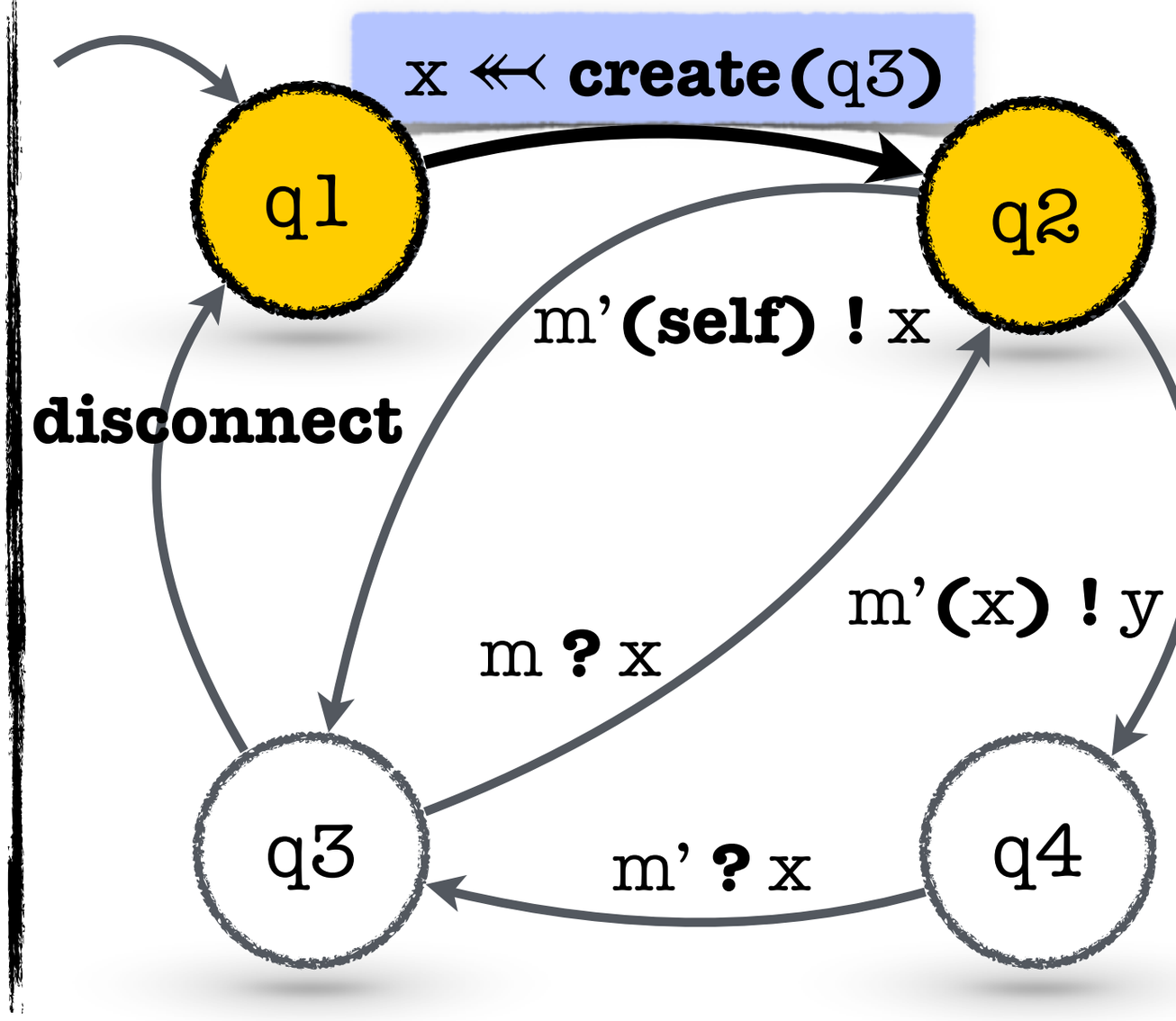


Dynamic Register Automata

Verification of Buffered Dynamic Register Automata

Process Creation

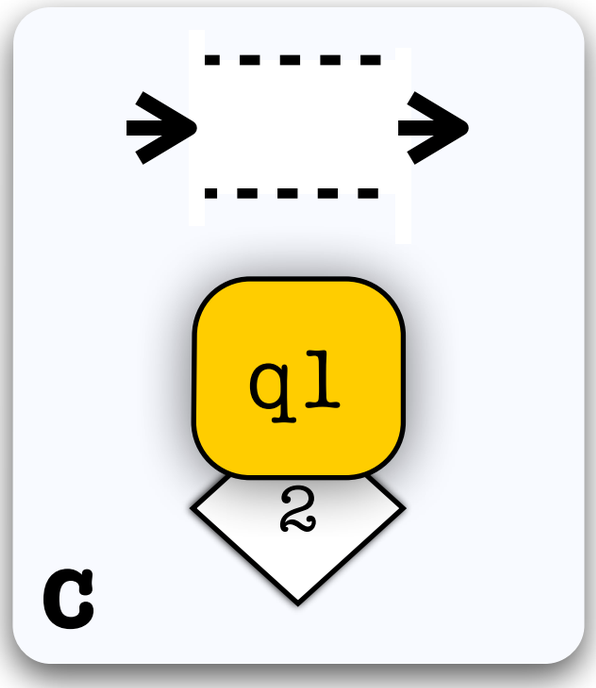


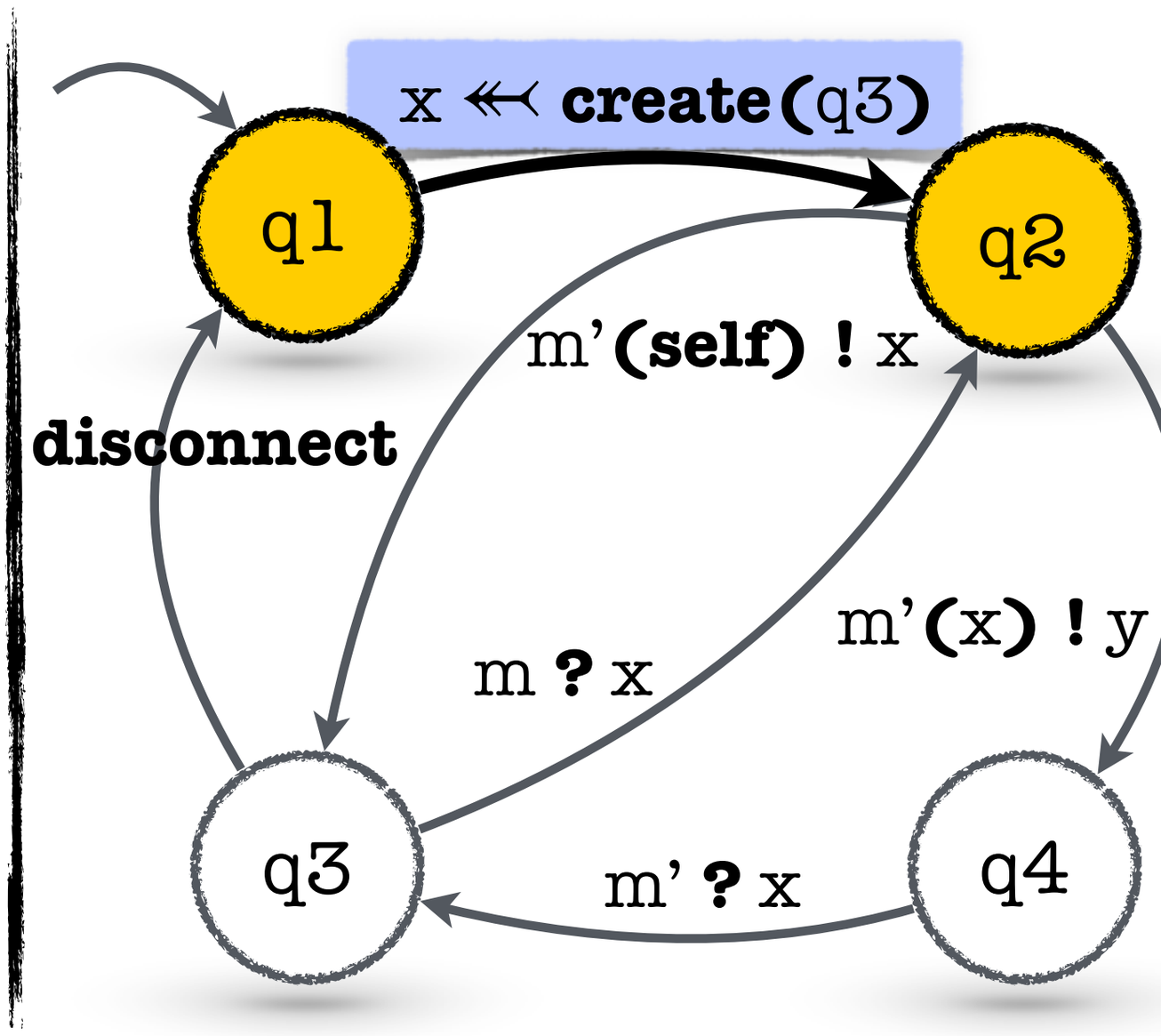


Dynamic Register Automata

Verification of Buffered Dynamic Register Automata

Process Creation

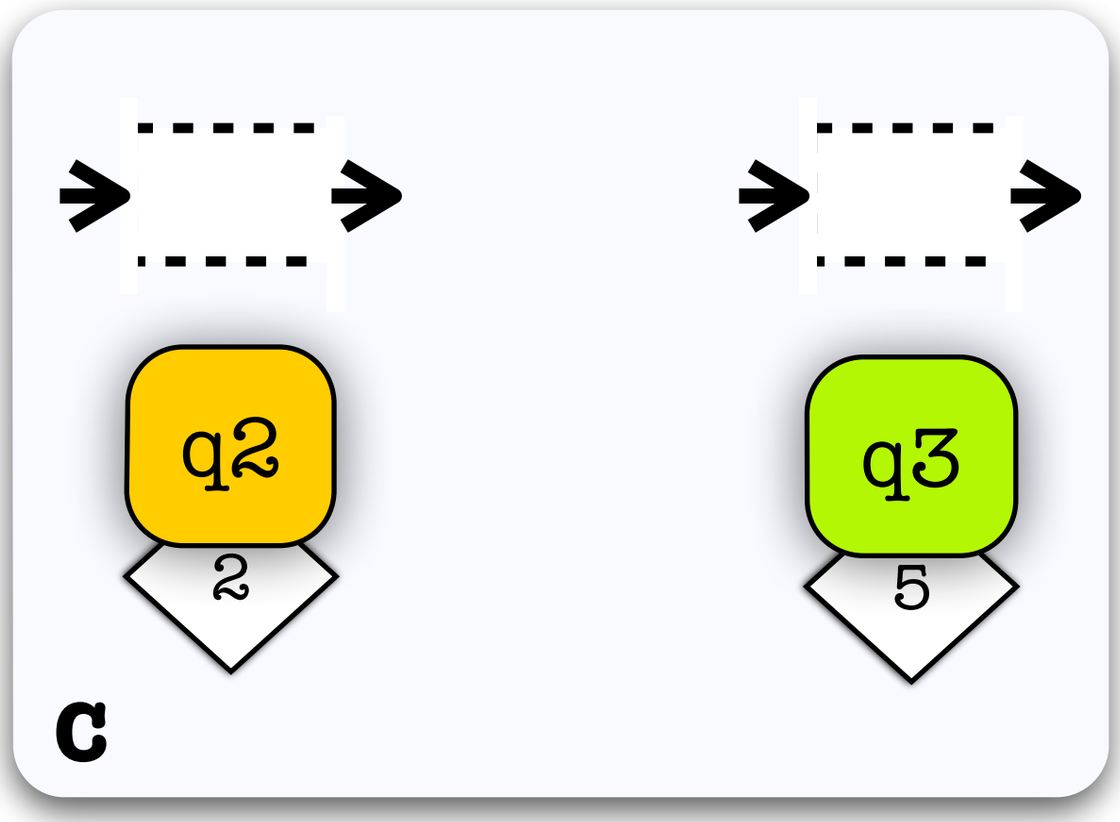


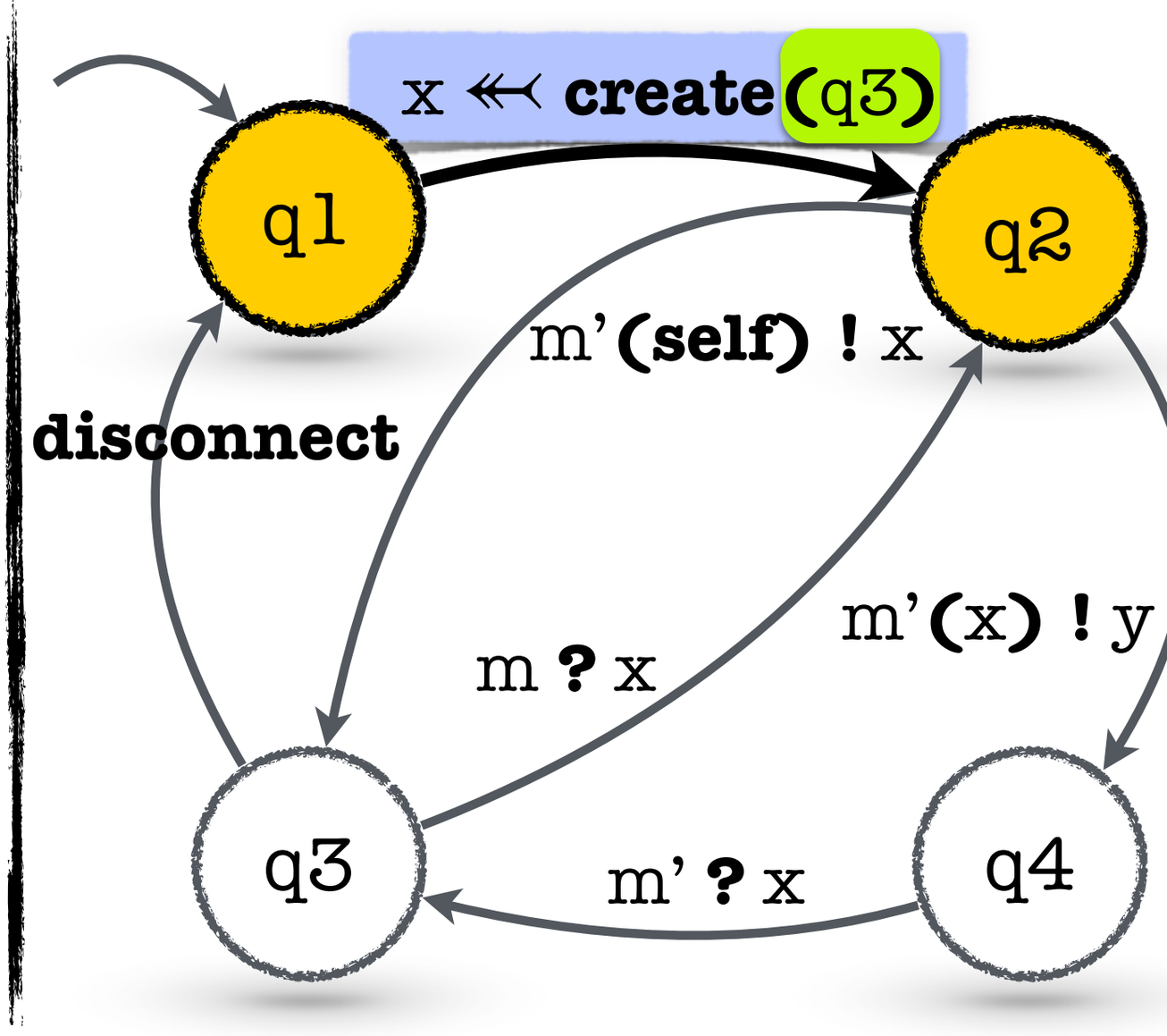


Dynamic Register Automata

Verification of Buffered Dynamic Register Automata

Process Creation

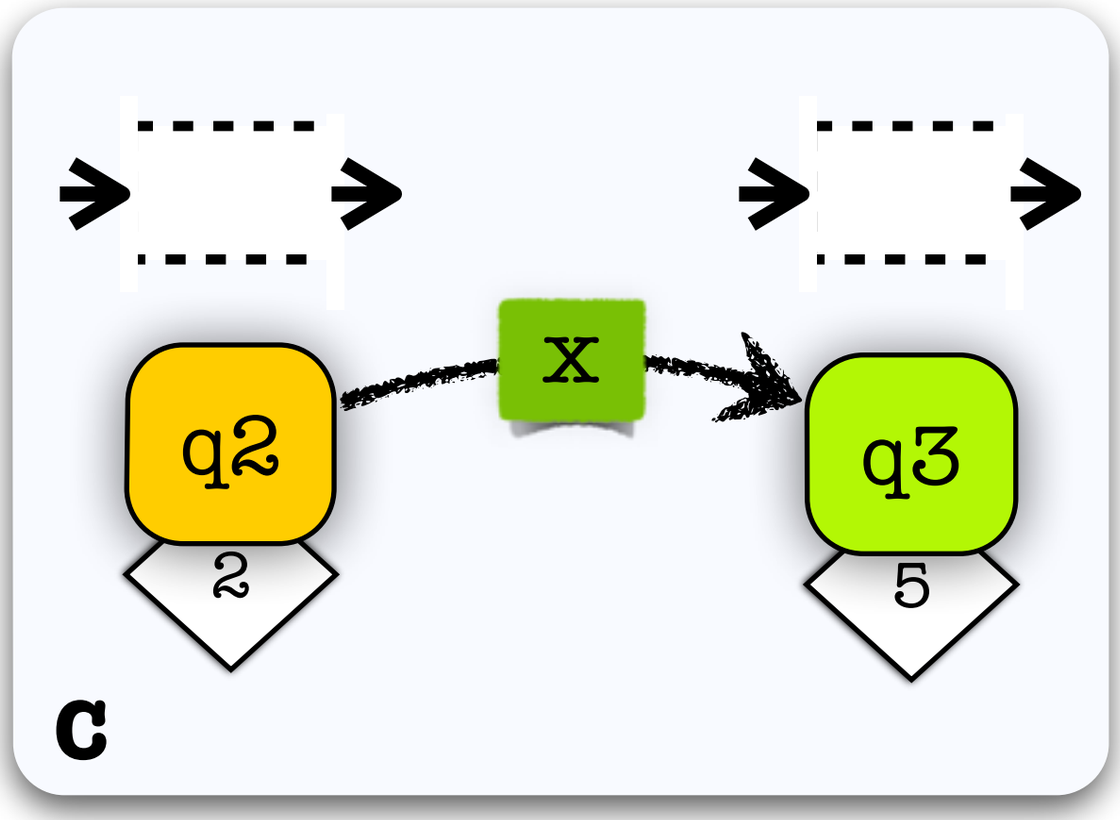


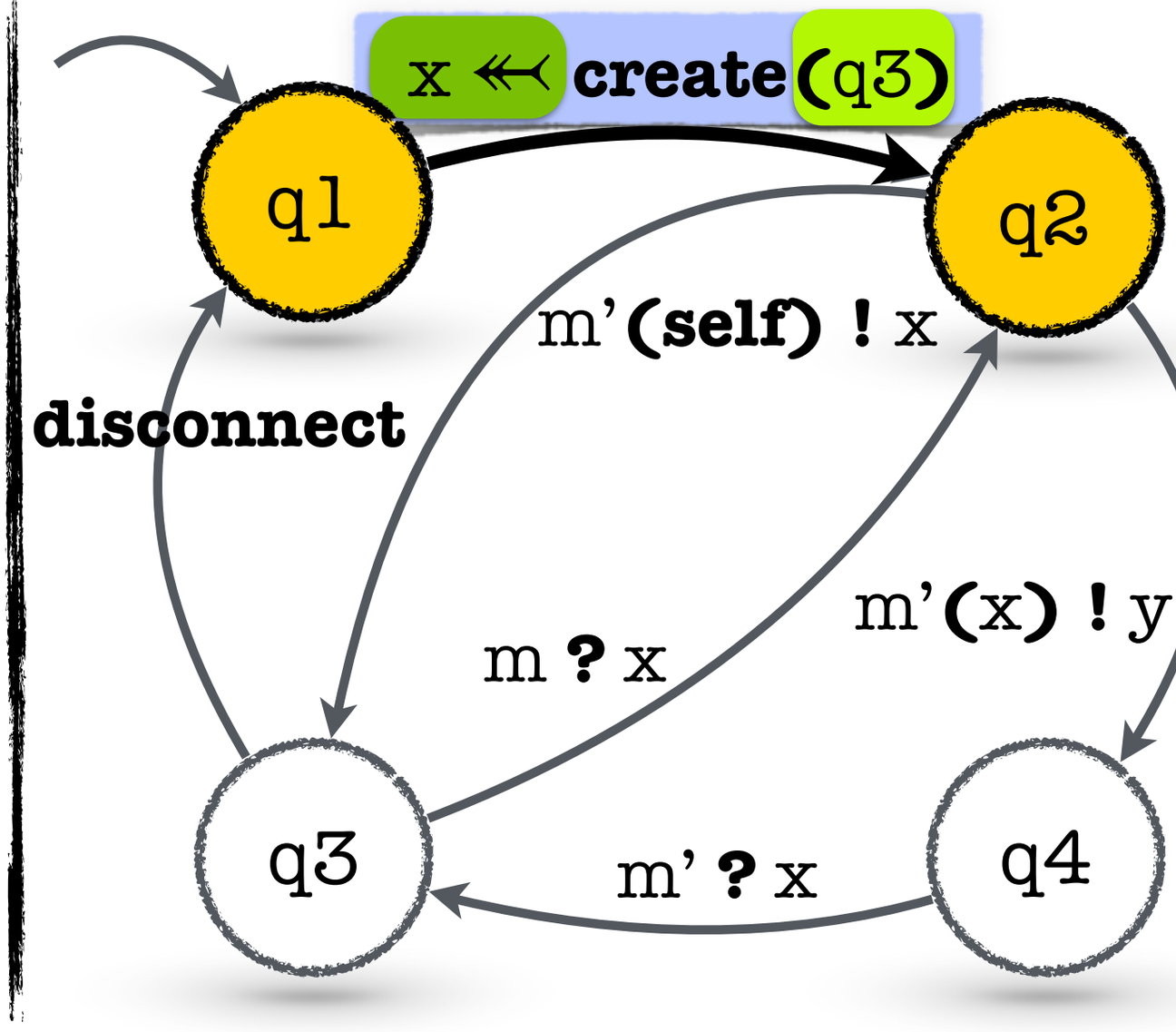


Dynamic Register Automata

Verification of Buffered Dynamic Register Automata

Process Creation

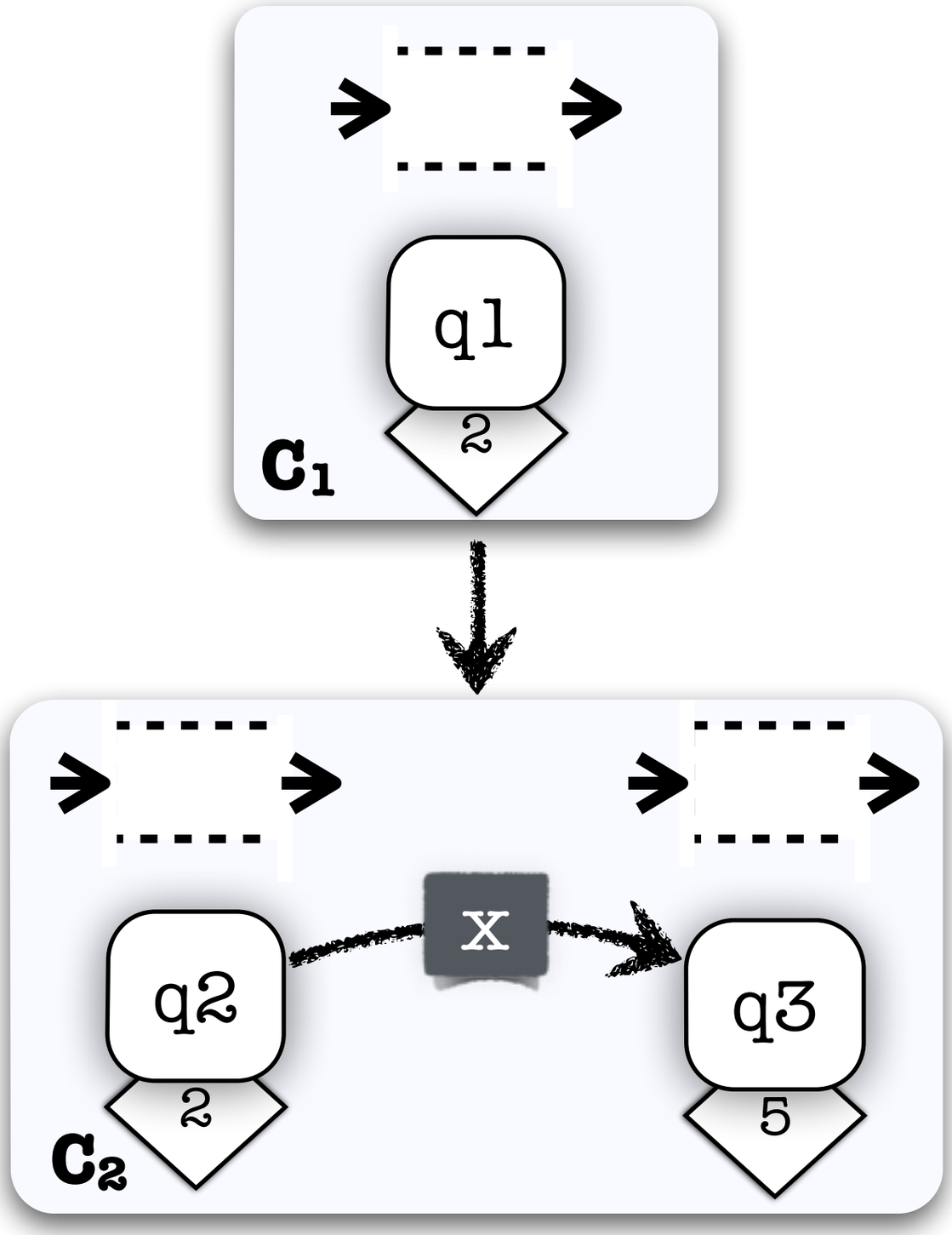


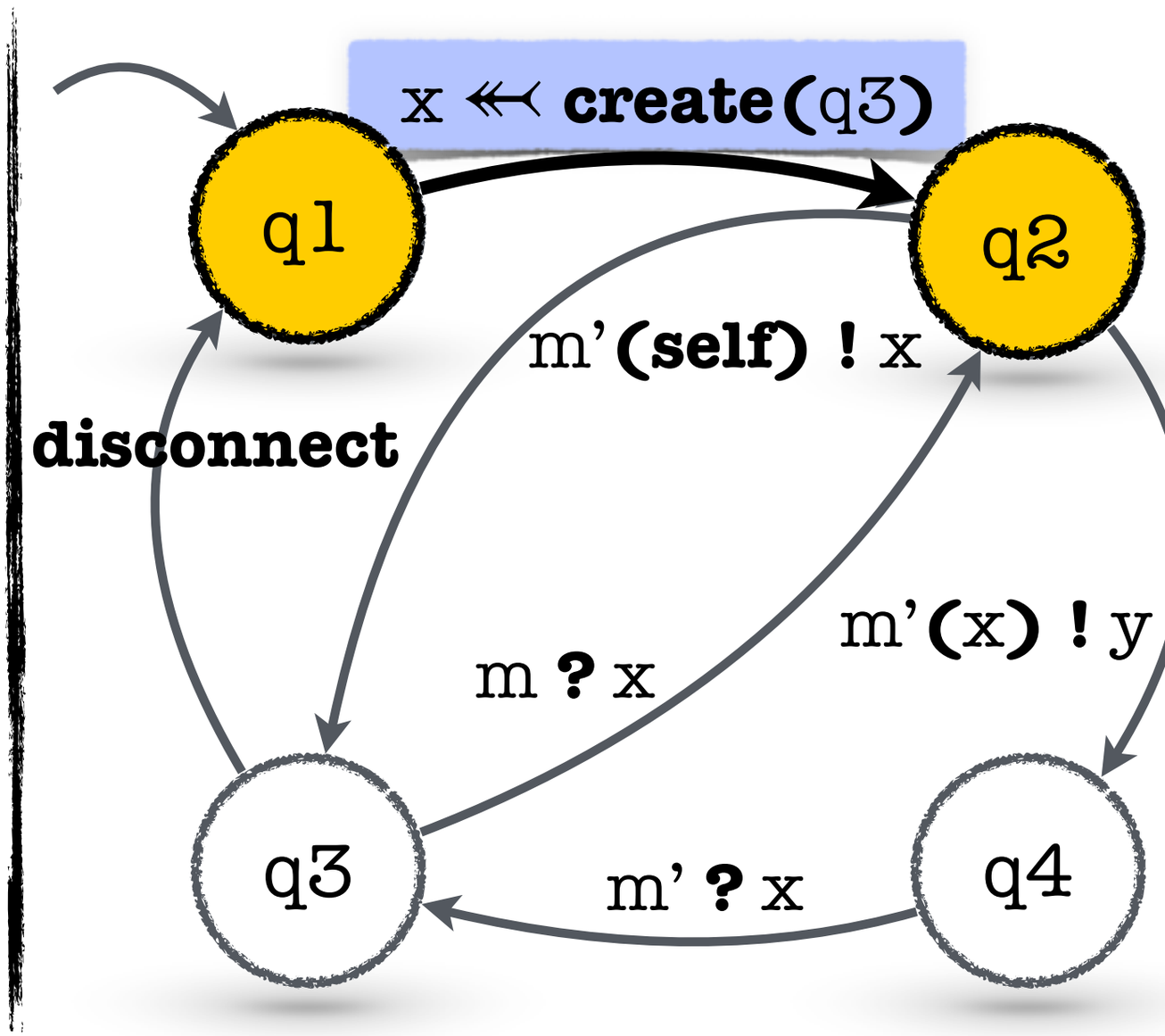


Dynamic Register Automata

Verification of Buffered Dynamic Register Automata

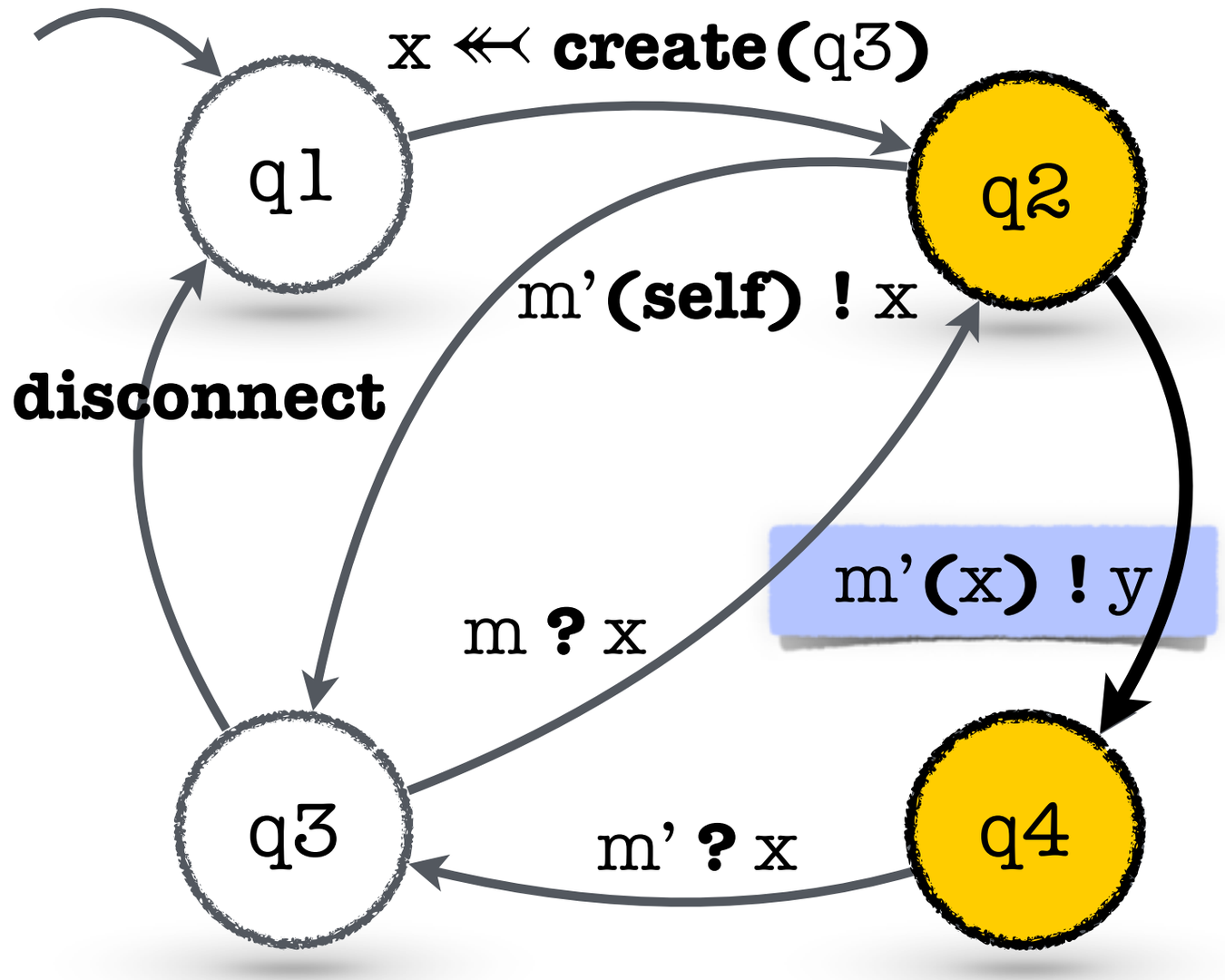
Process Creation





Operational Semantics

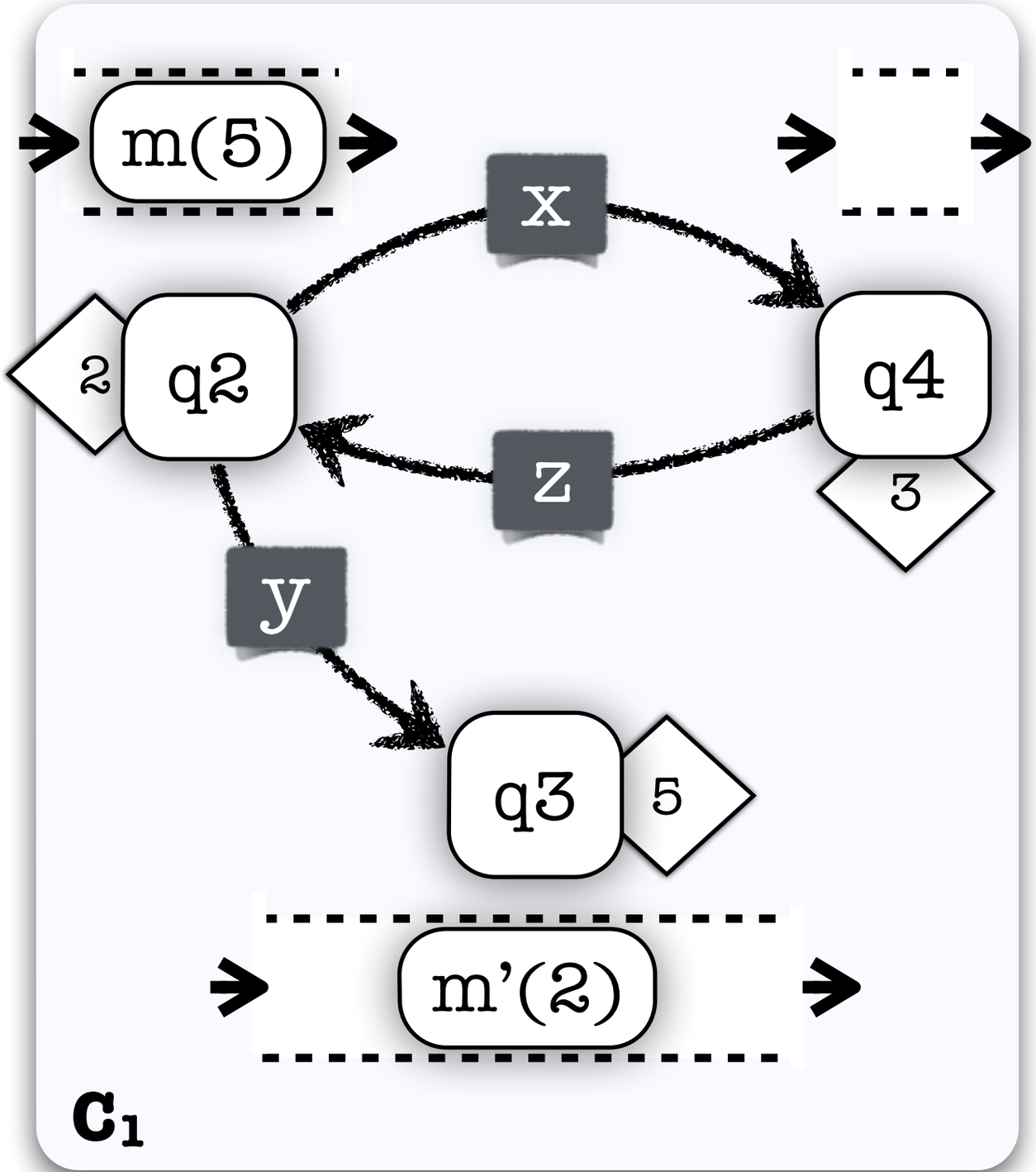
Process Creation **Message Sending**

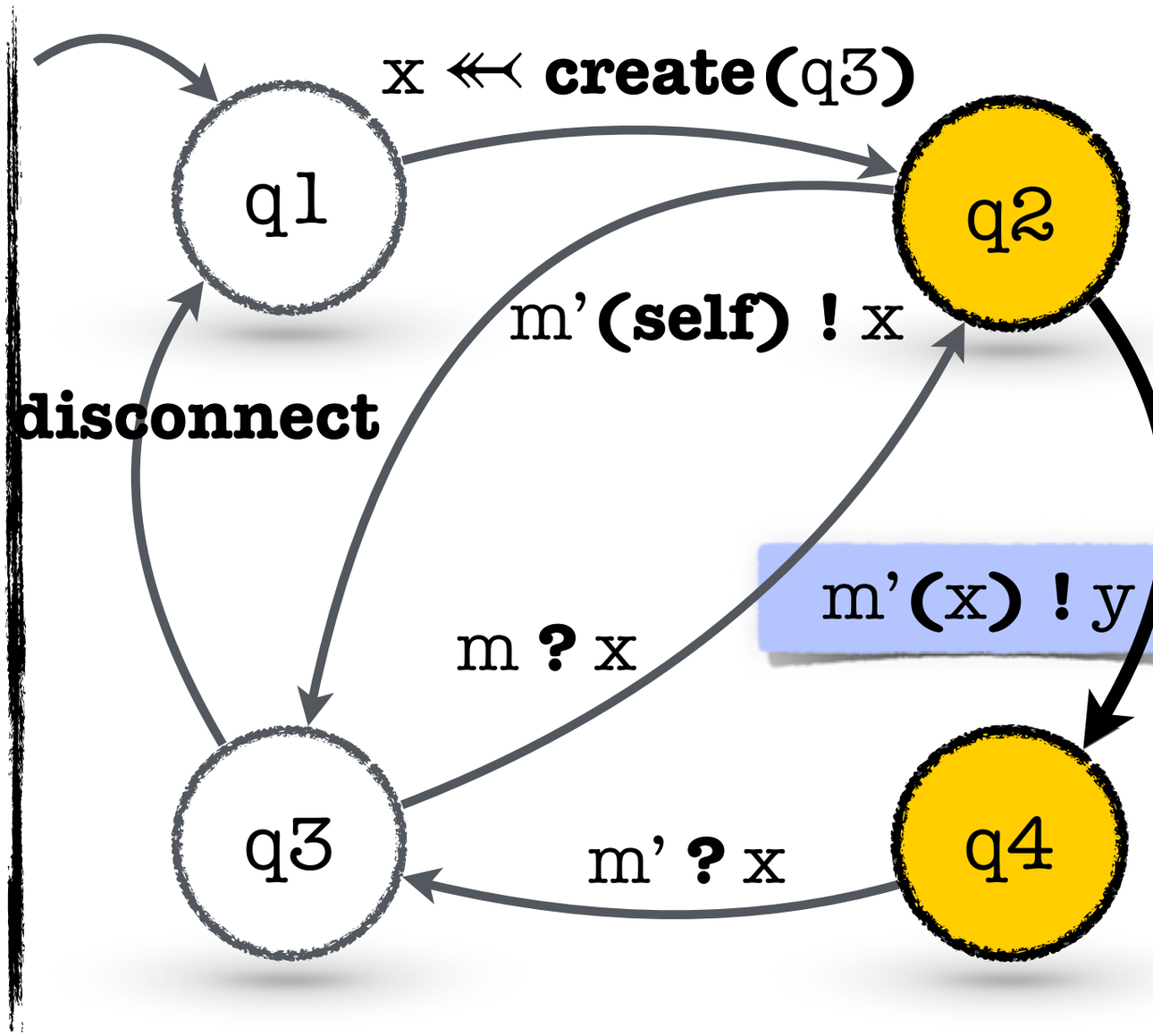


Operational Semantics

Process Creation

Message Sending

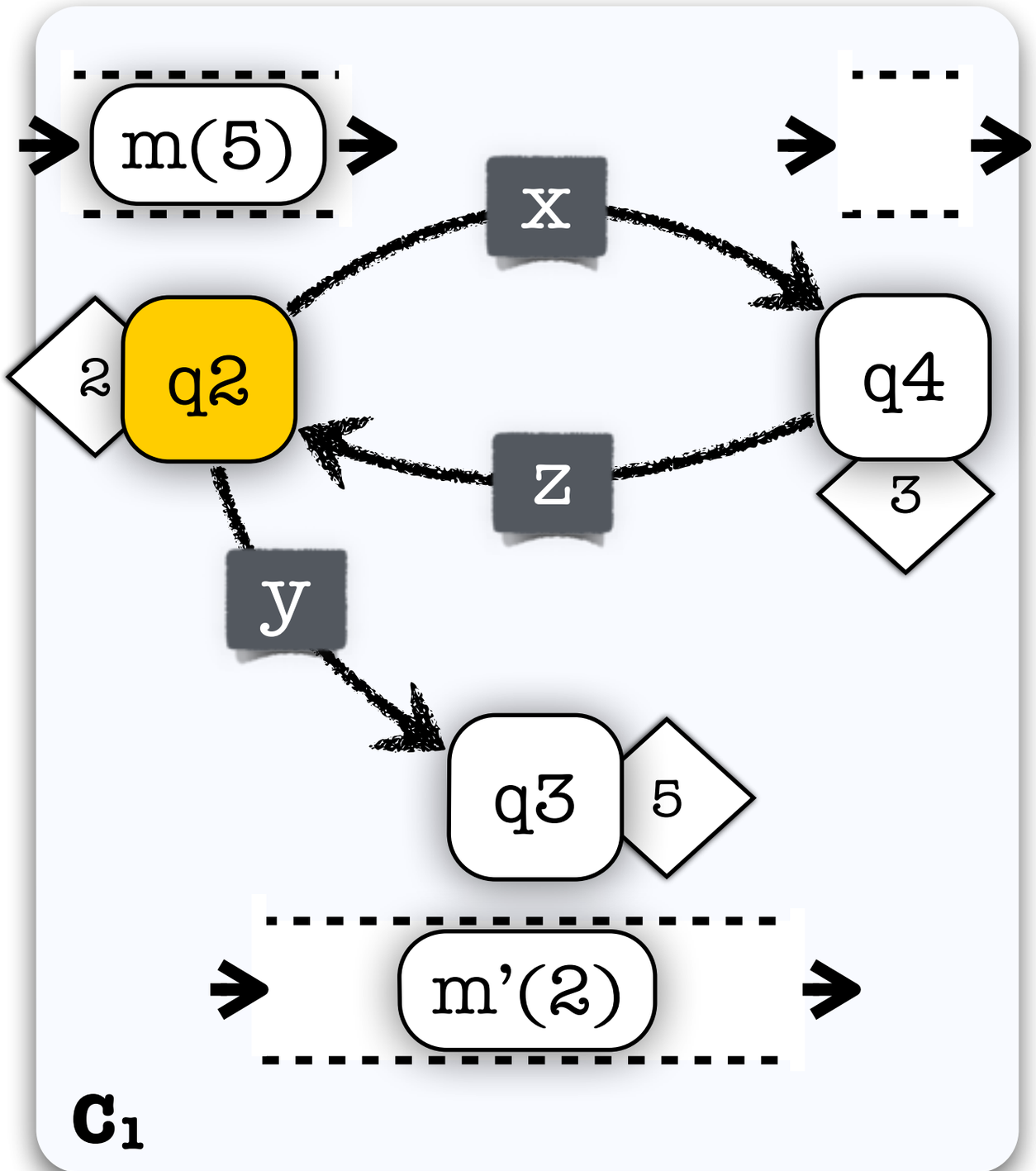




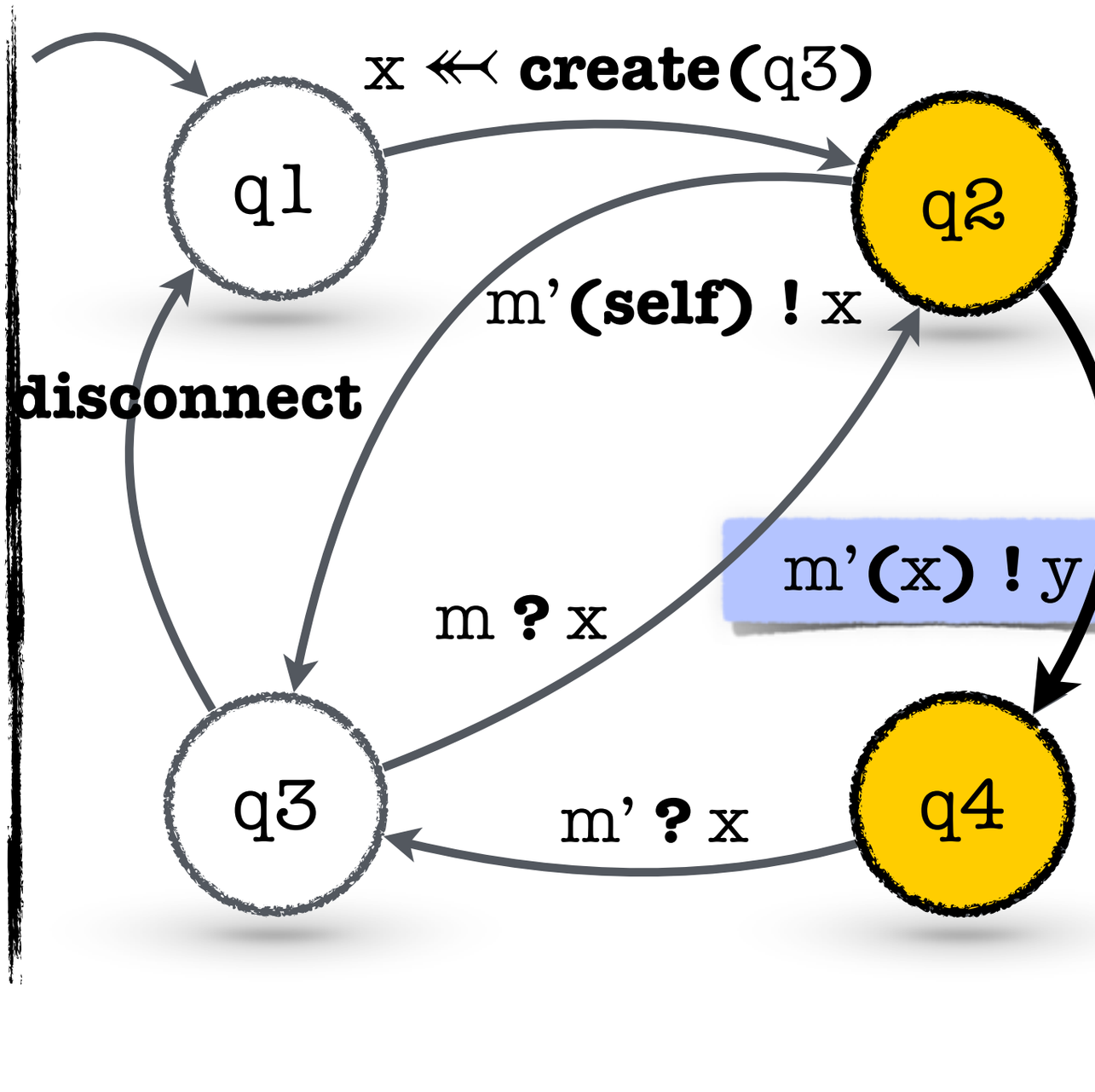
Operational Semantics

Verification of Buffered Dynamic Register Automata

Process Creation



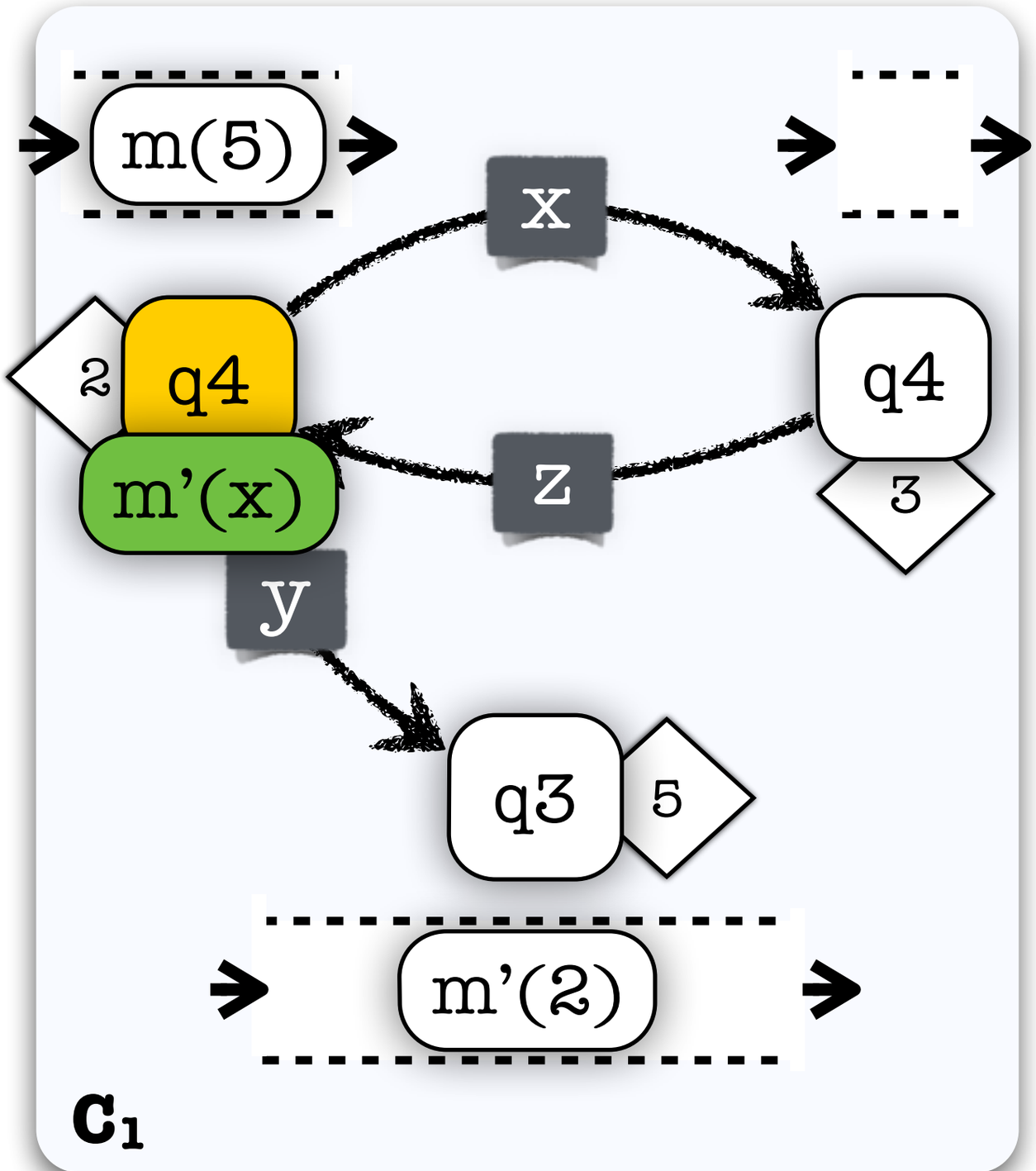
Message Sending



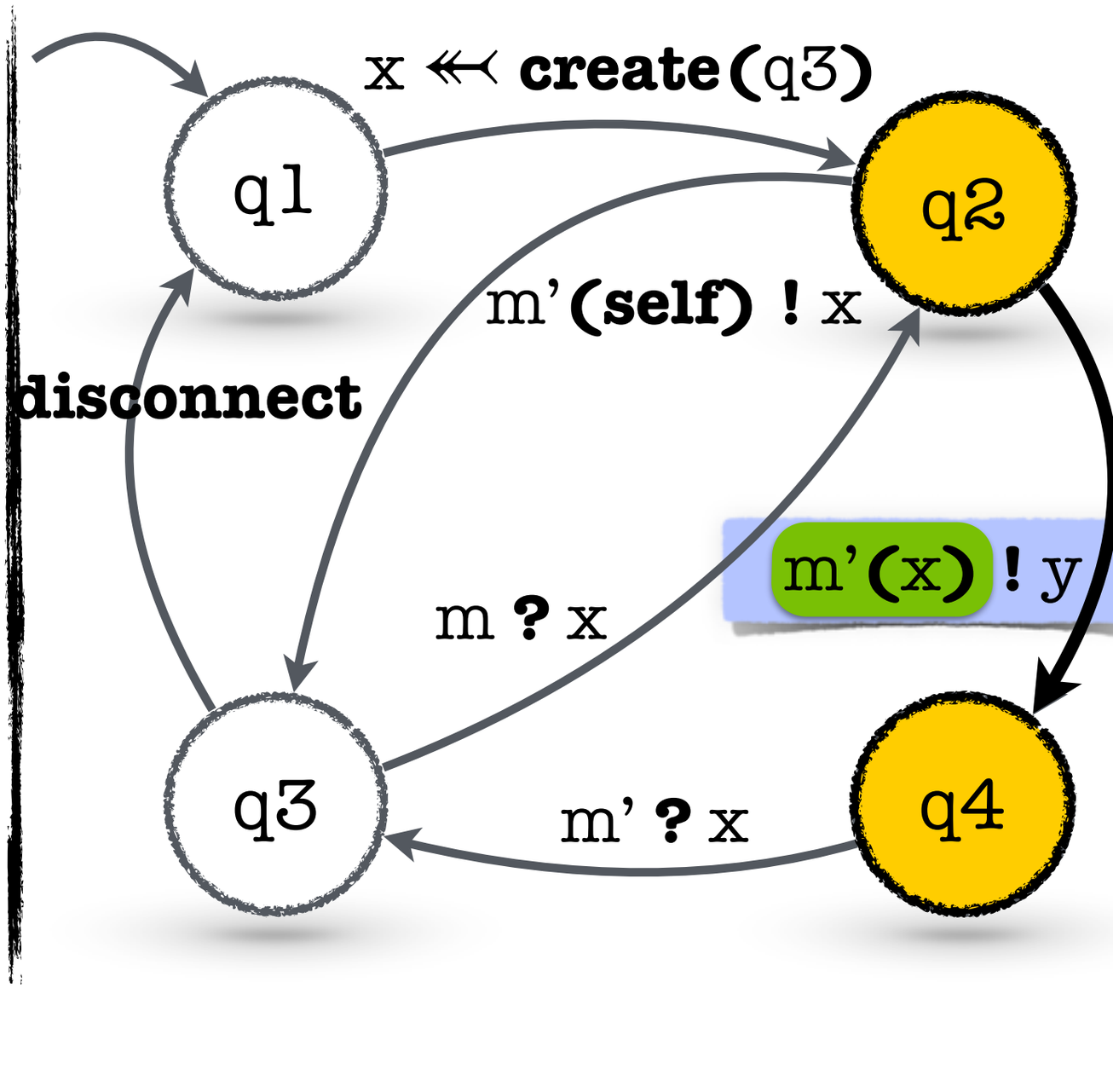
Operational Semantics

Verification of Buffered Dynamic Register Automata

Process Creation



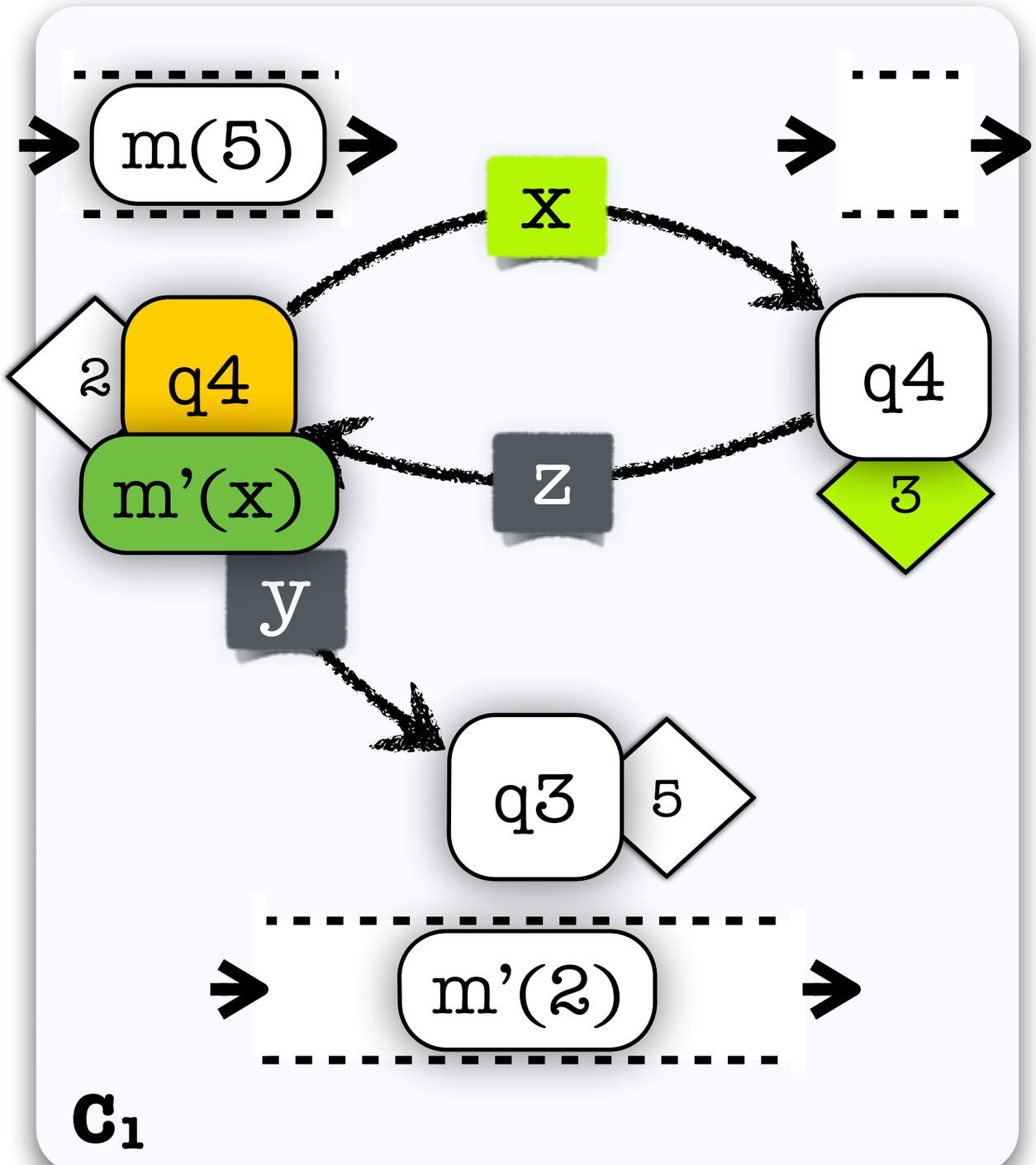
Message Sending



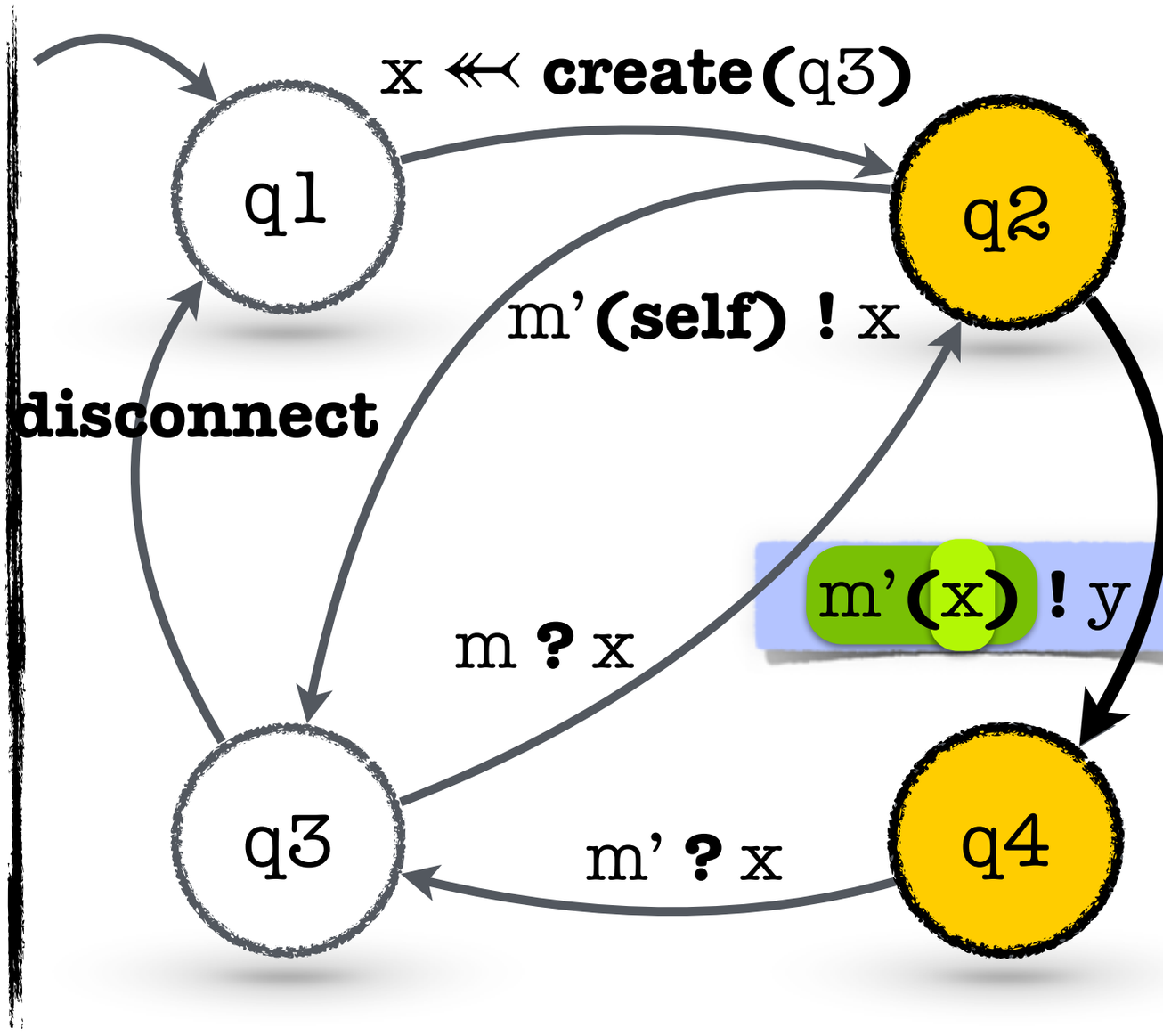
Operational Semantics

Verification of Buffered Dynamic Register Automata

Process Creation



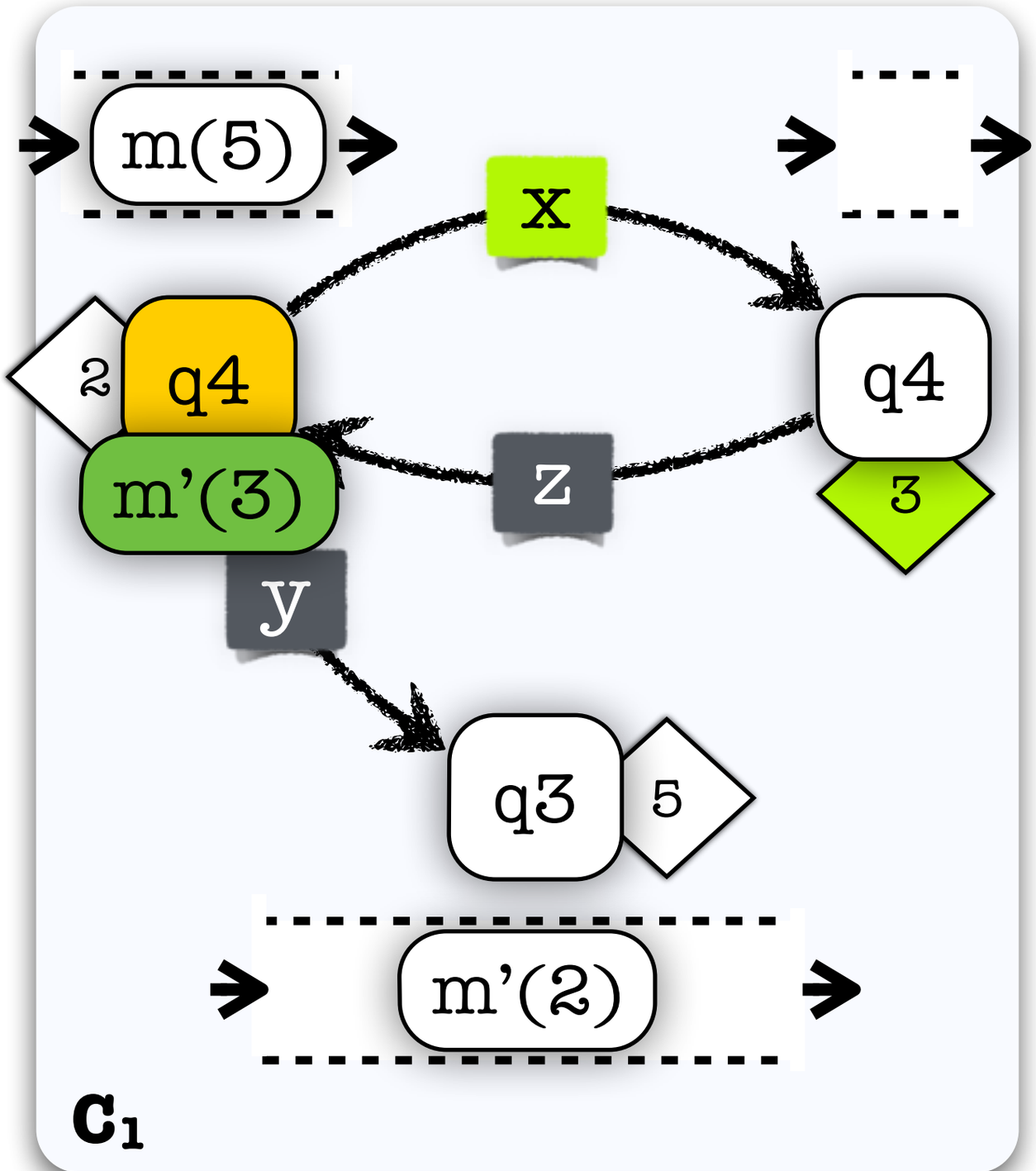
Message Sending



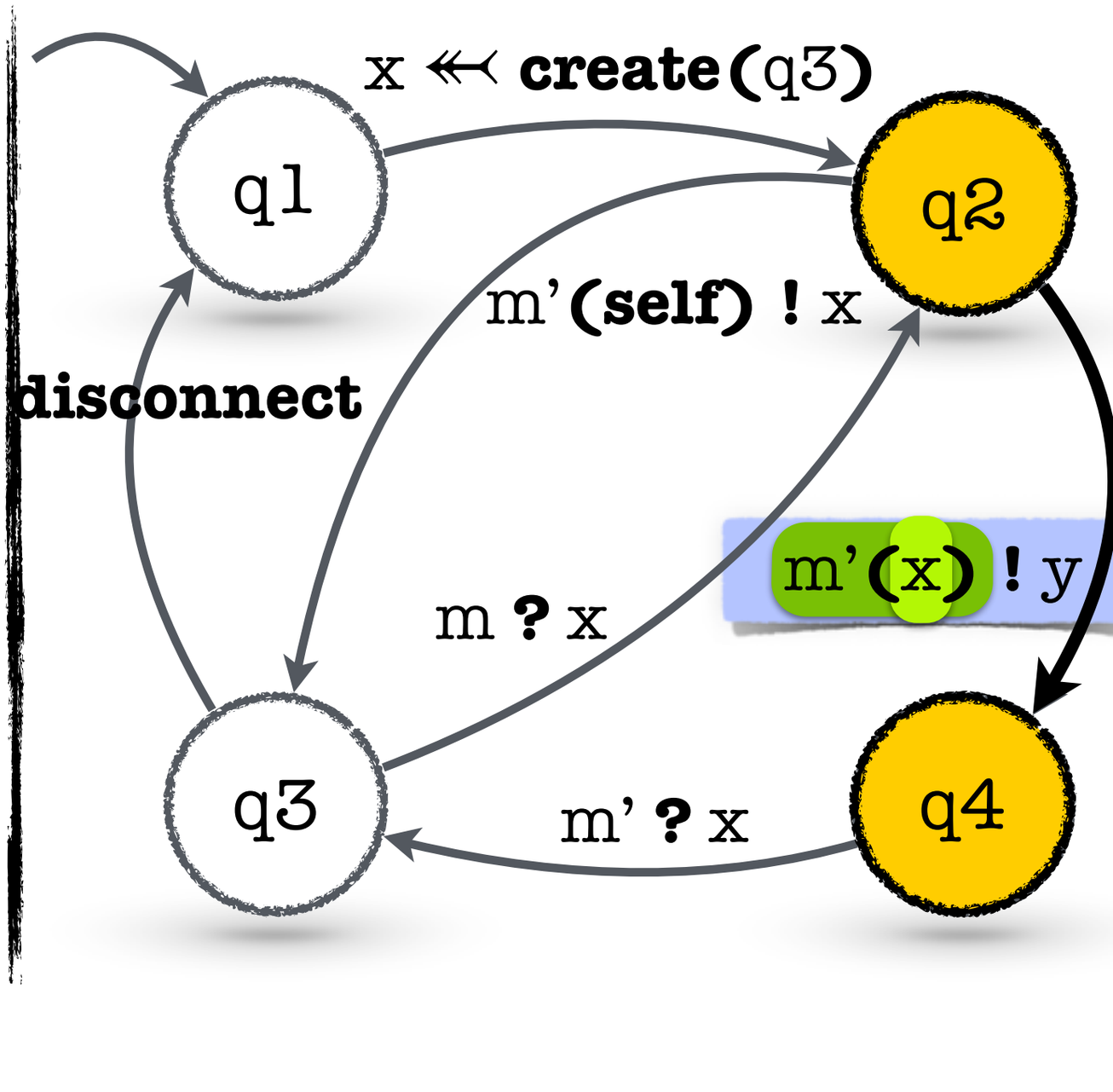
Operational Semantics

Verification of Buffered Dynamic Register Automata

Process Creation



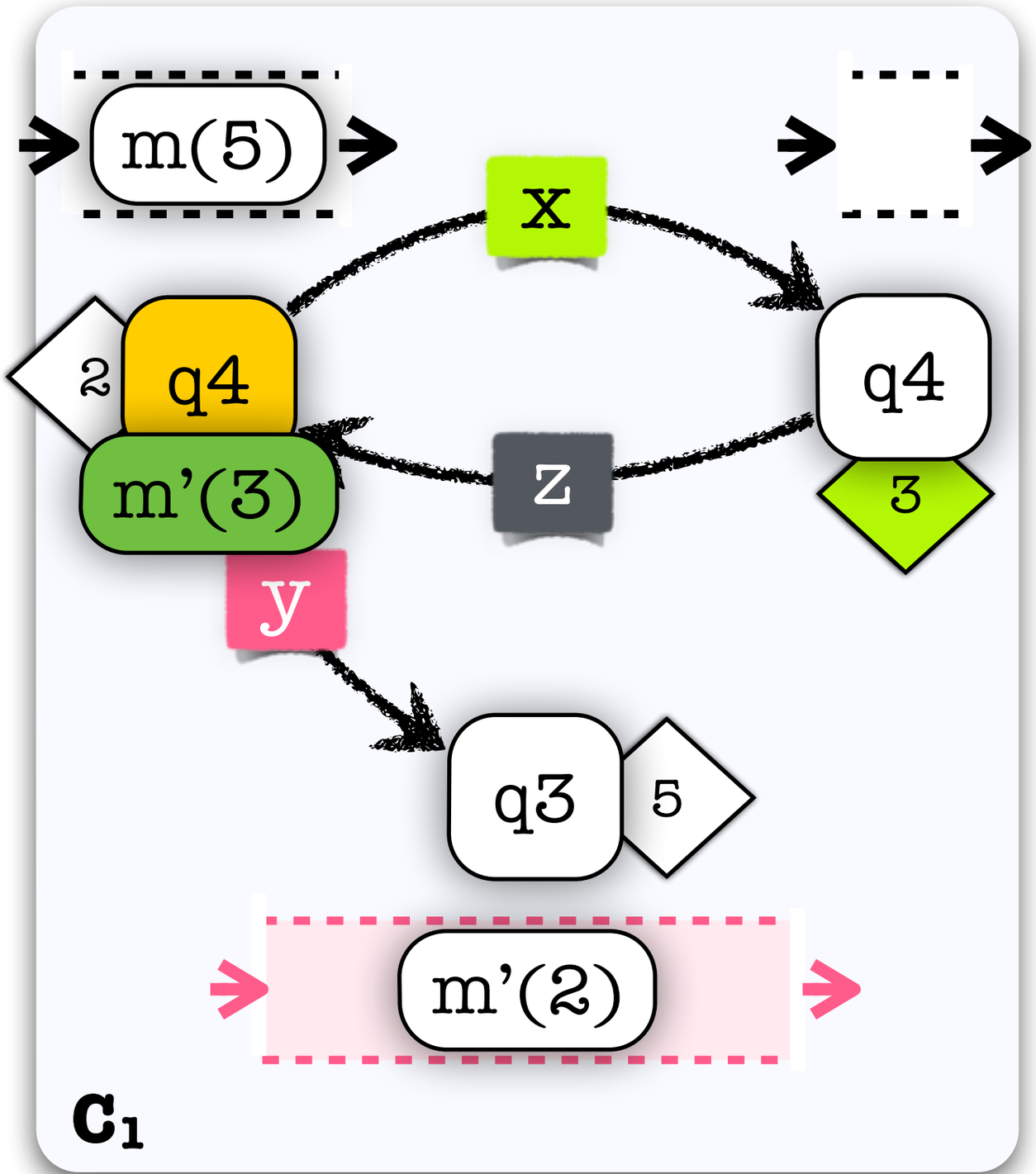
Message Sending



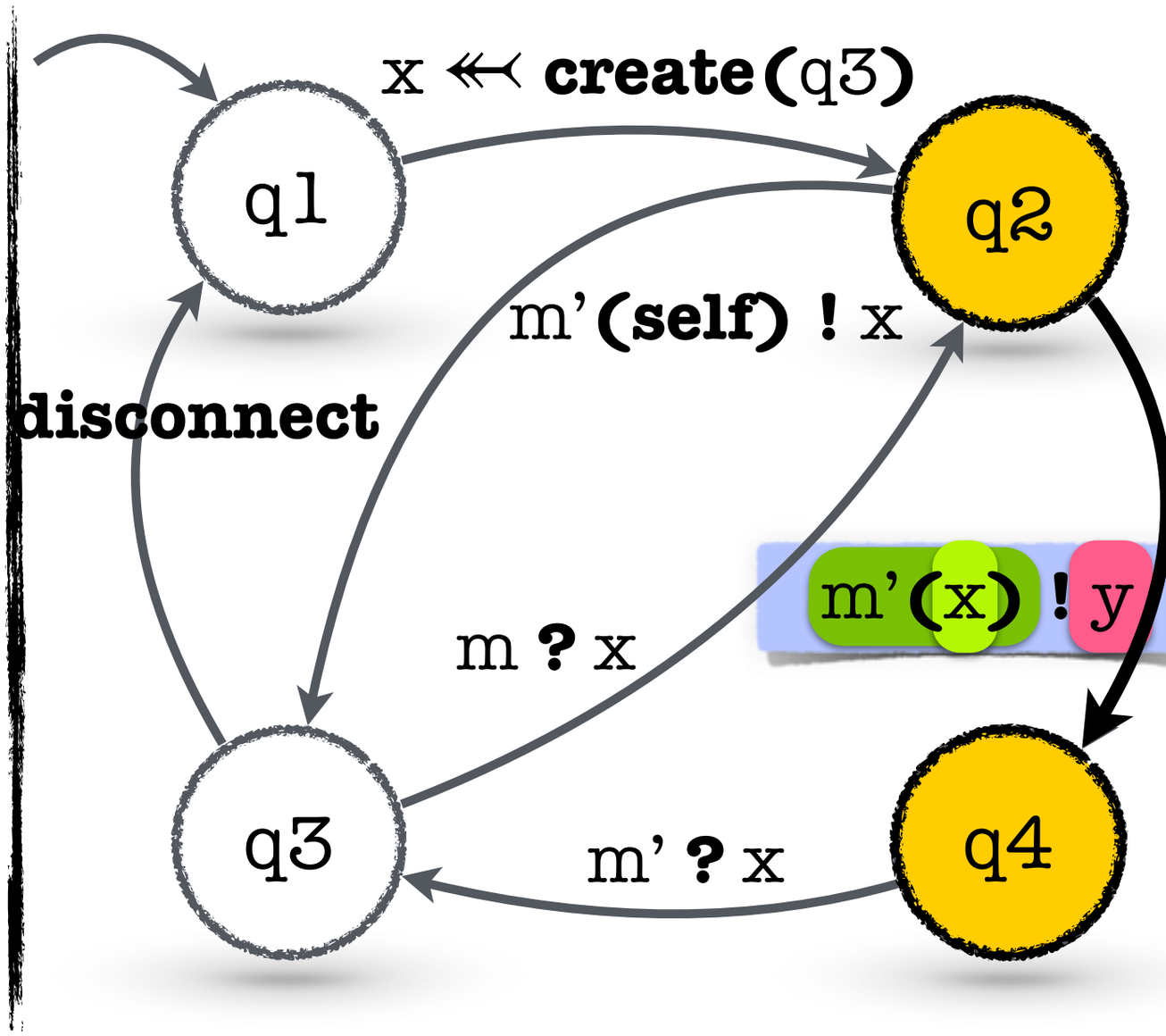
Operational Semantics

Verification of Buffered Dynamic Register Automata

Process Creation



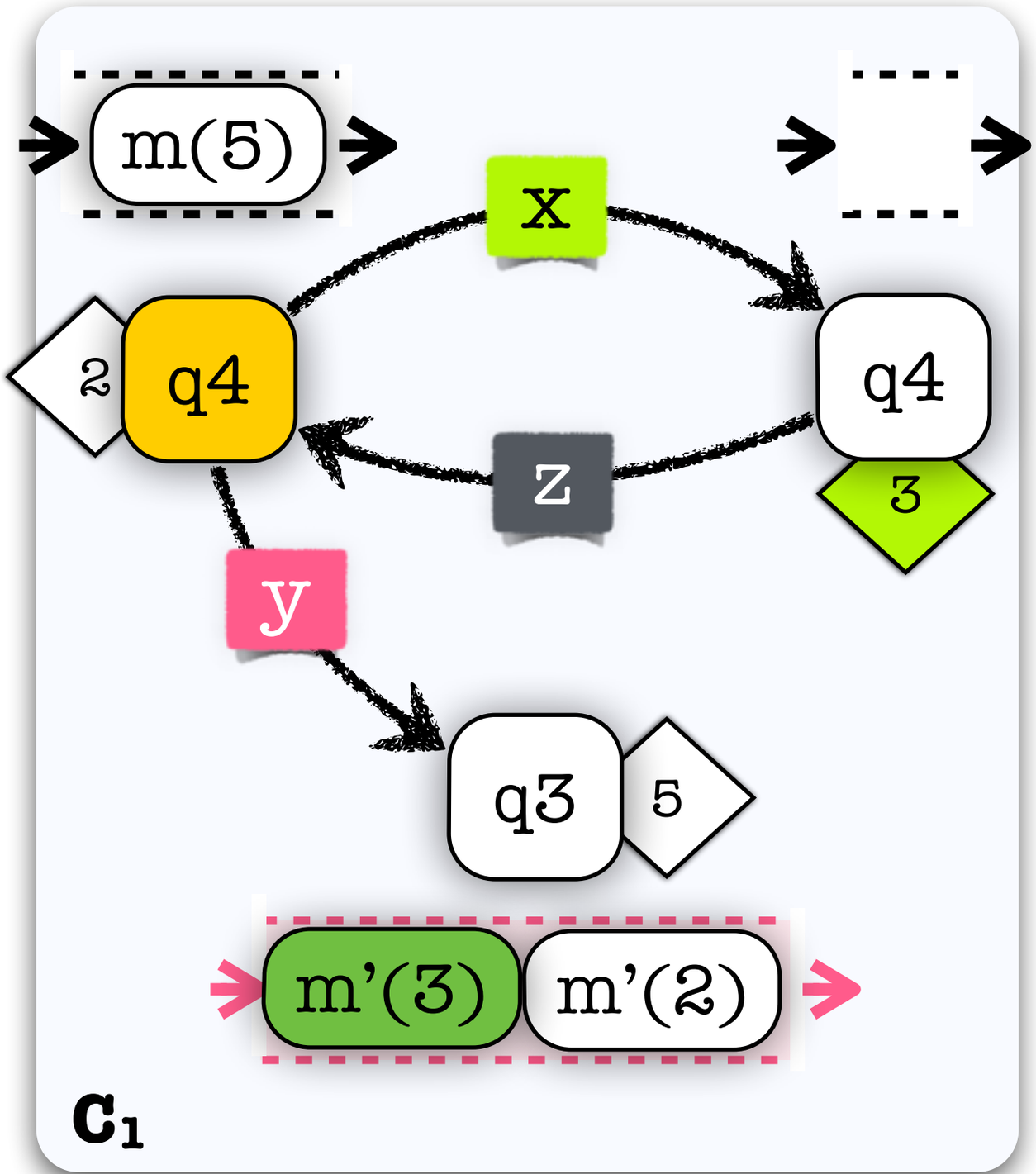
Message Sending



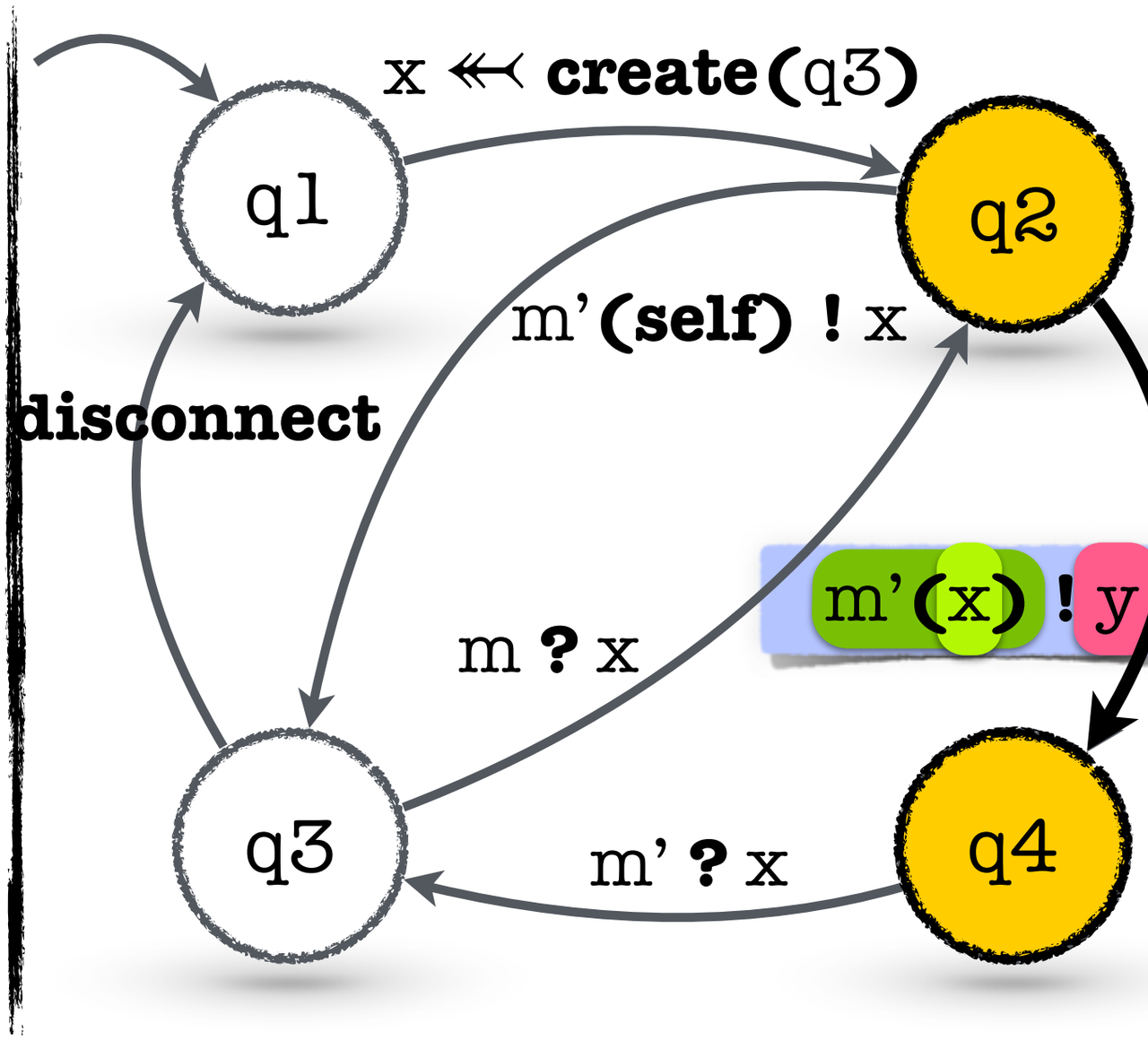
Operational Semantics

Verification of Buffered Dynamic Register Automata

Process Creation



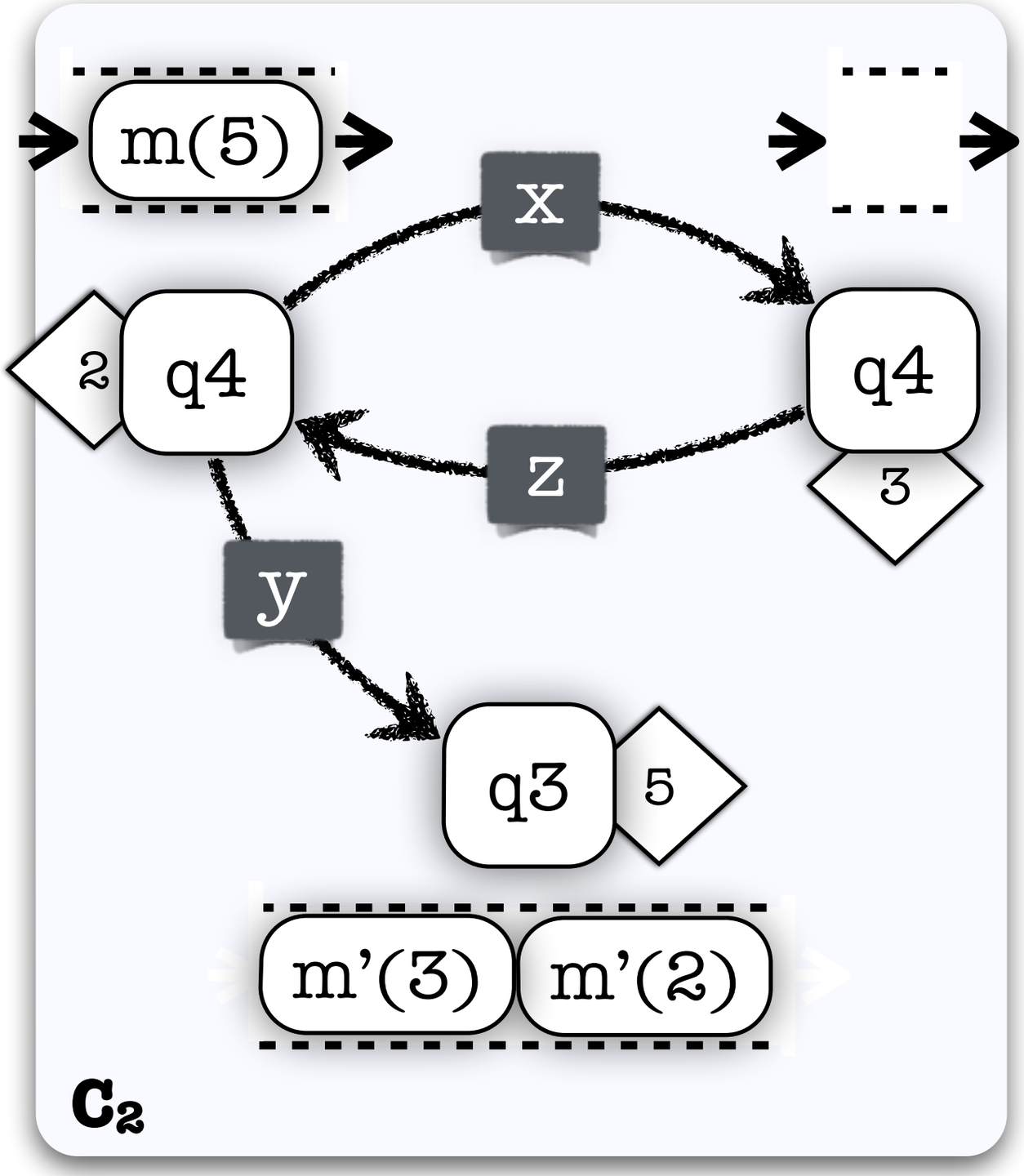
Message Sending



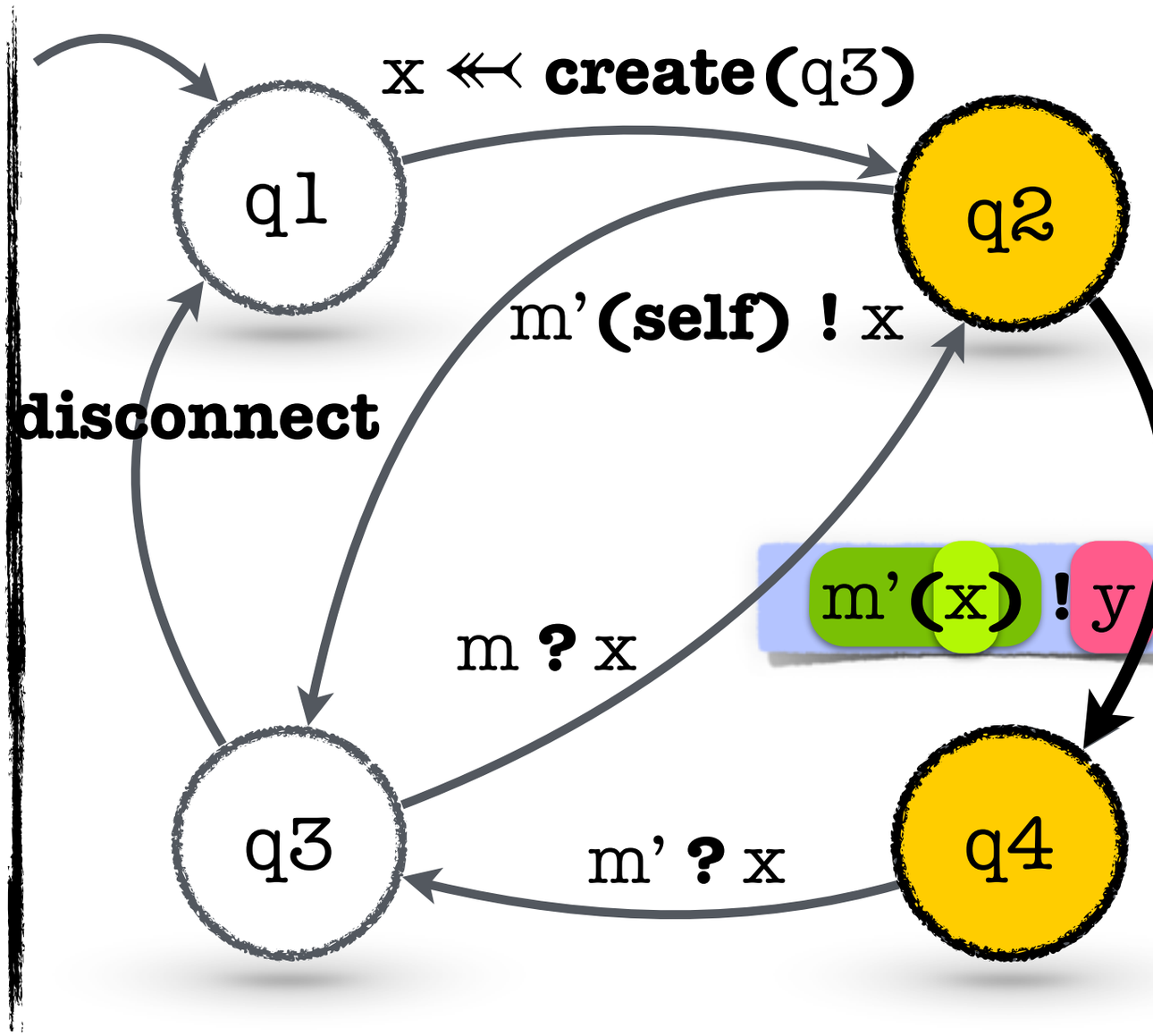
Operational Semantics

Verification of Buffered Dynamic Register Automata

Process Creation



Message Sending

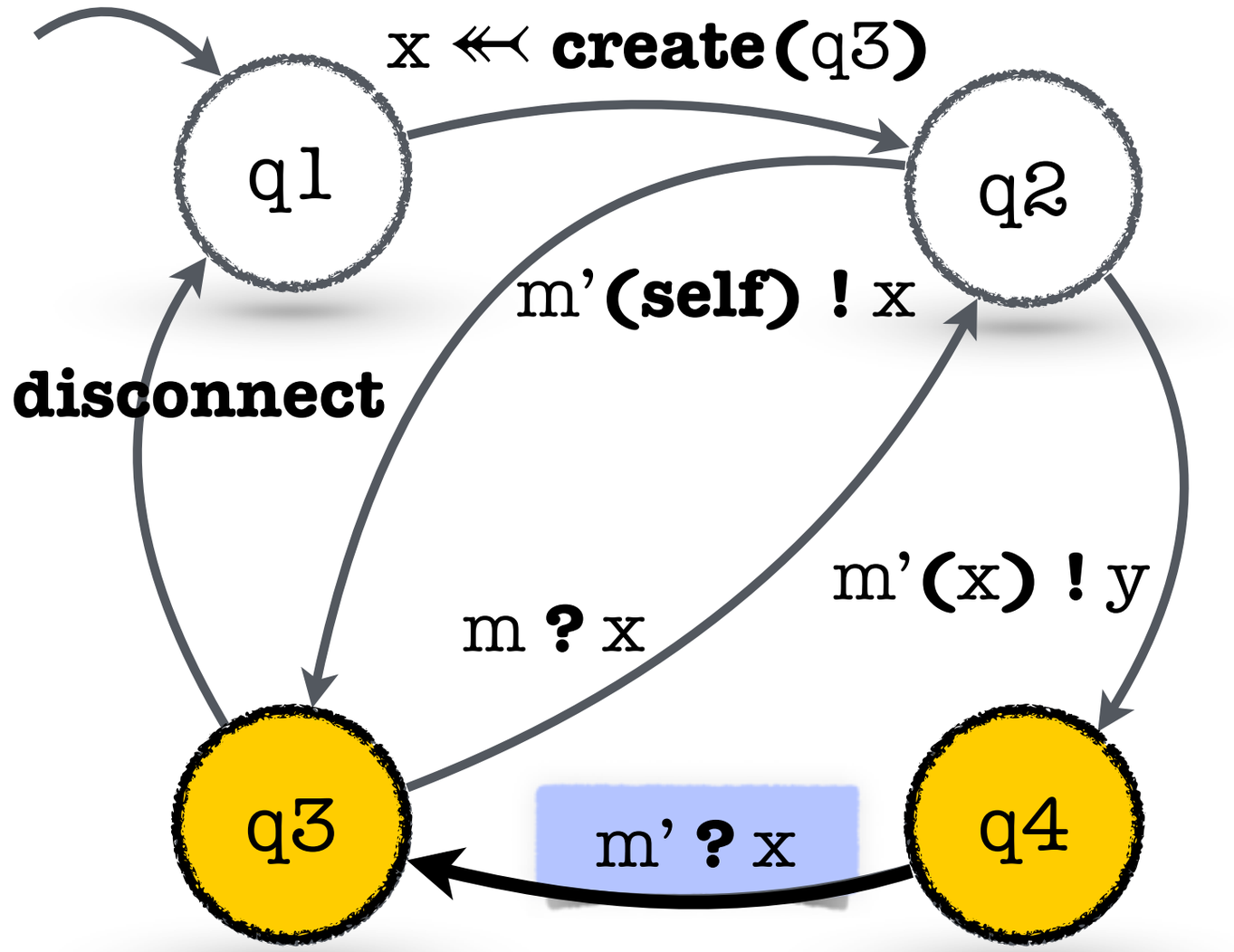


Operational Semantics

Verification of Buffered Dynamic Register Automata

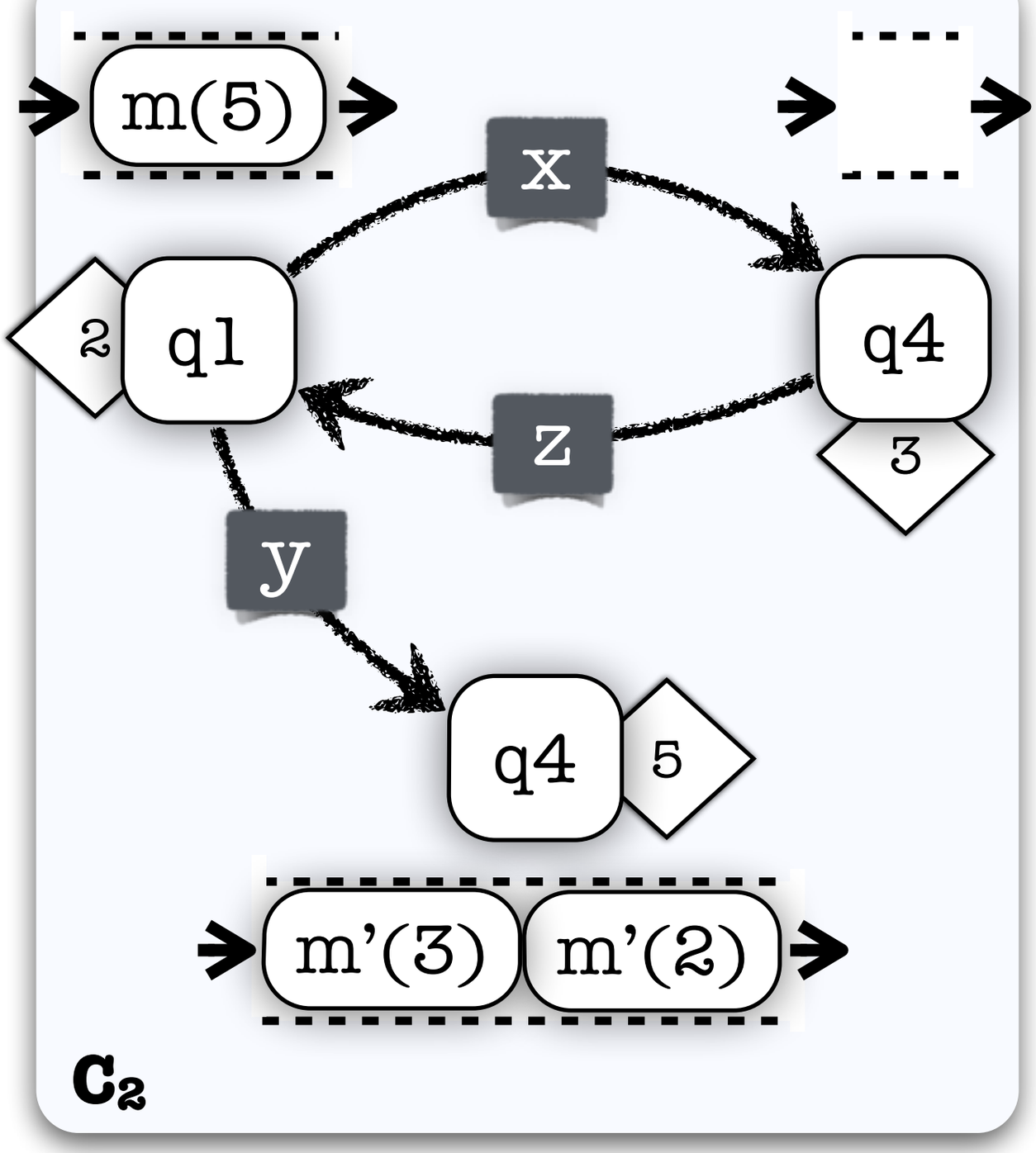
Message Sending

Message Receiving

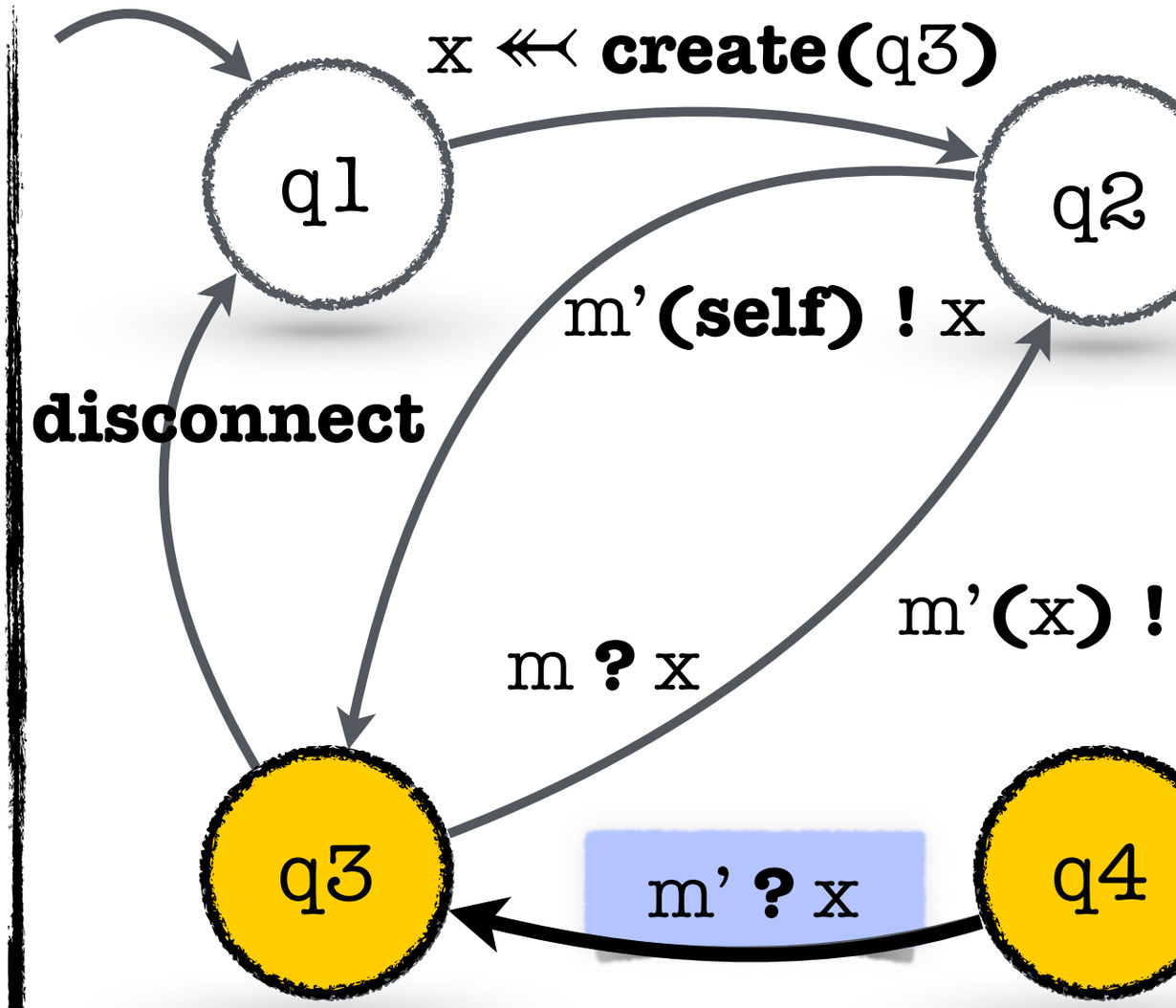


Operational Semantics

Message Sending



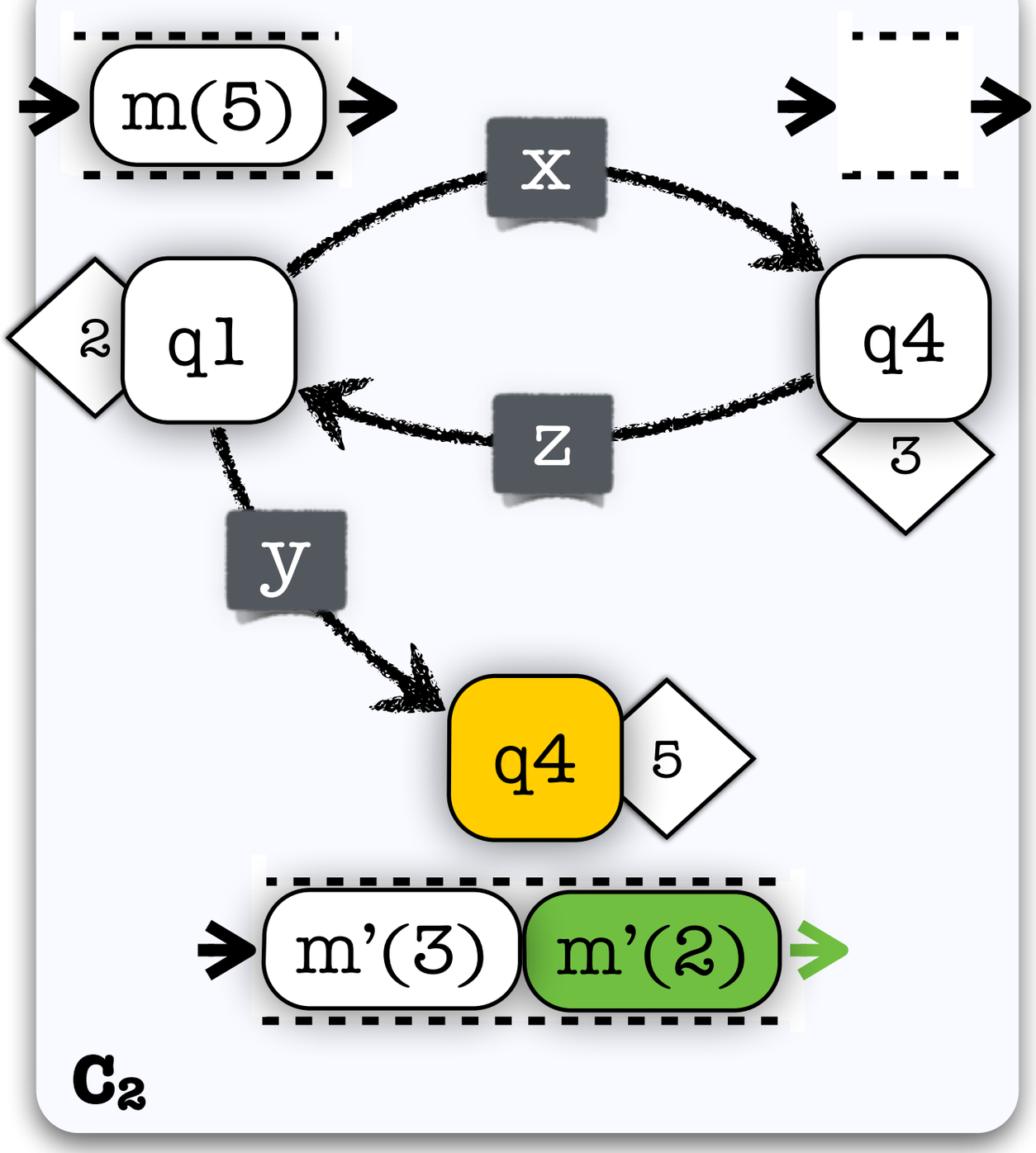
Message Receiving



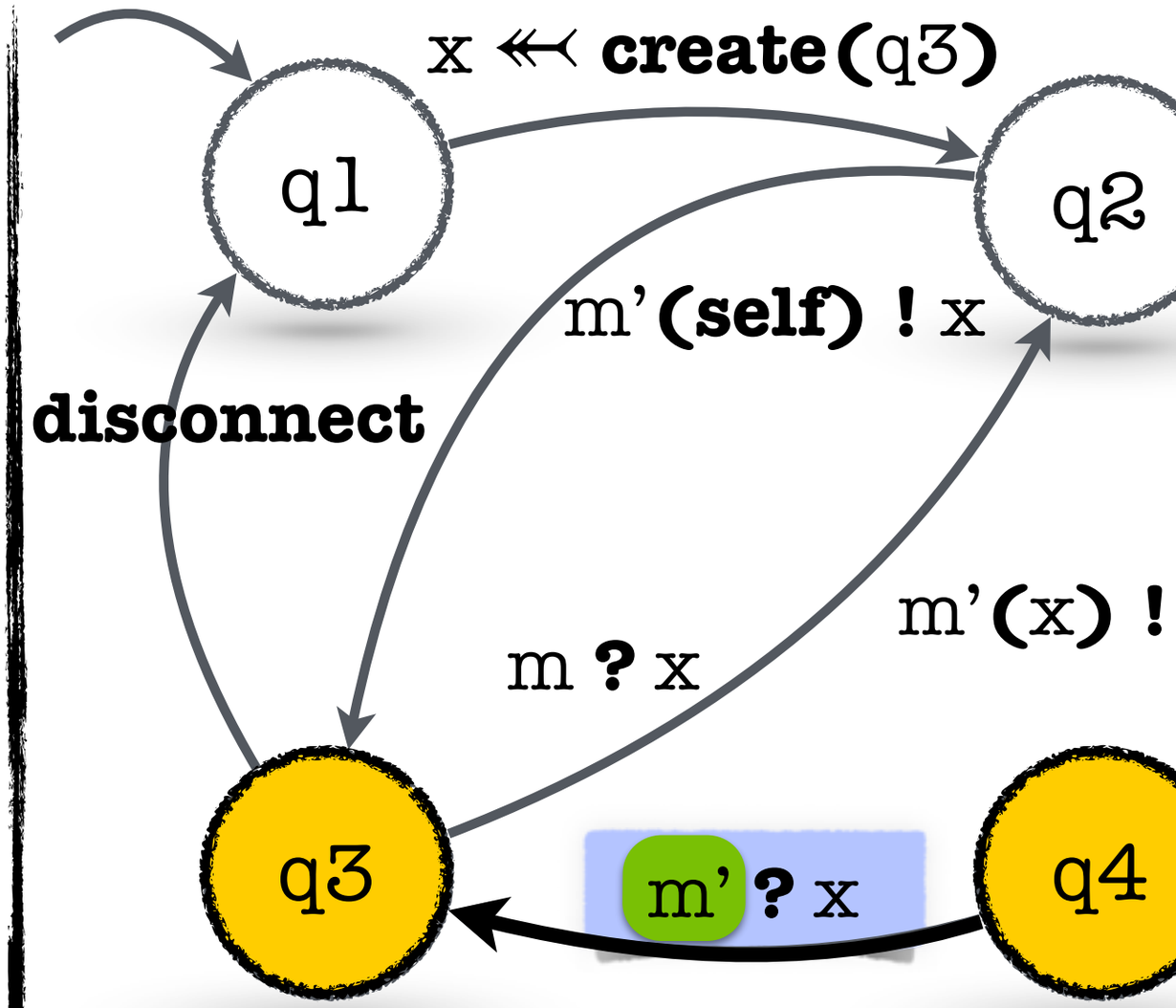
Operational Semantics

Verification of Buffered Dynamic Register Automata

Message Sending



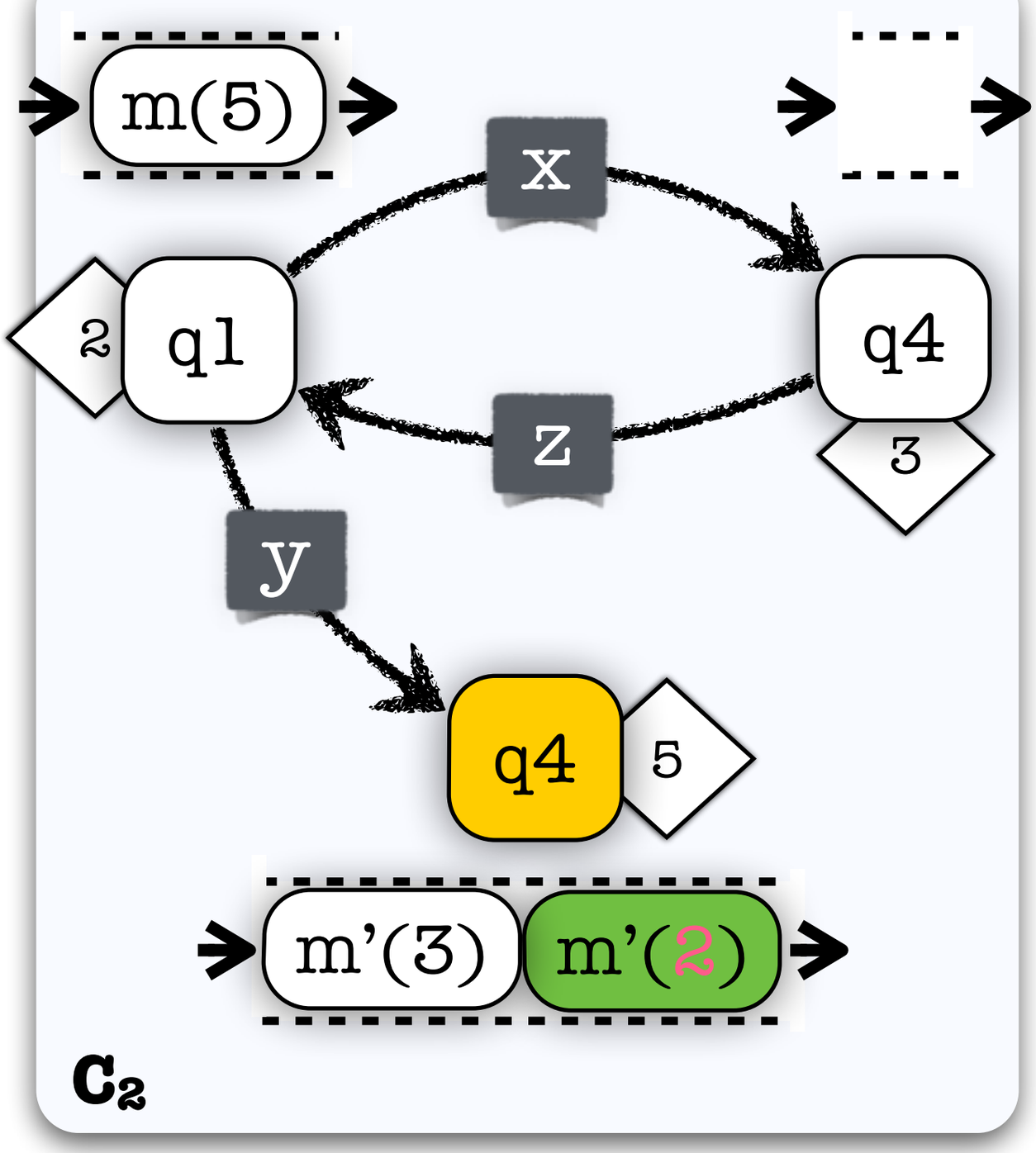
Message Receiving



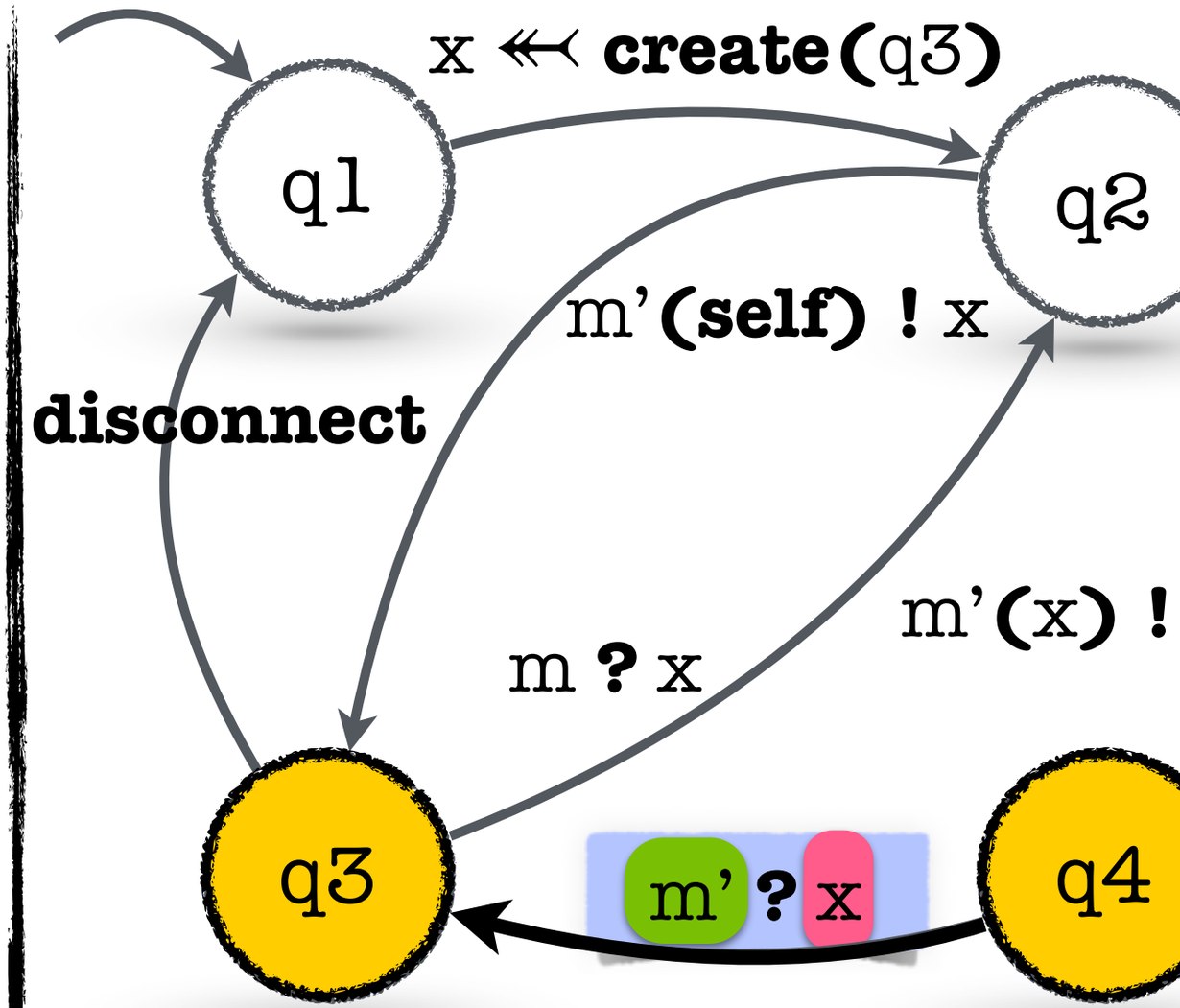
Operational Semantics

Verification of Buffered Dynamic Register Automata

Message Sending



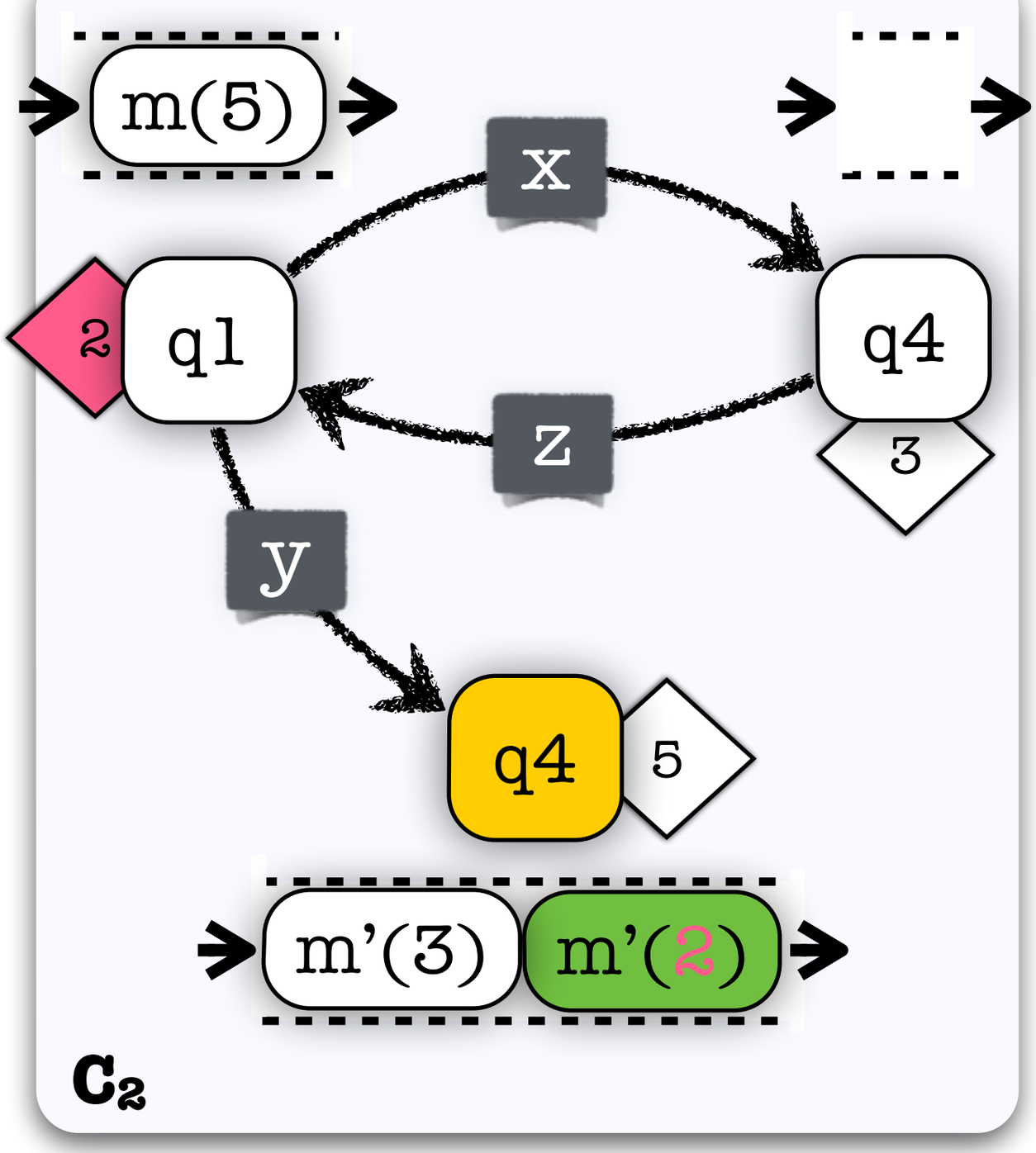
Message Receiving



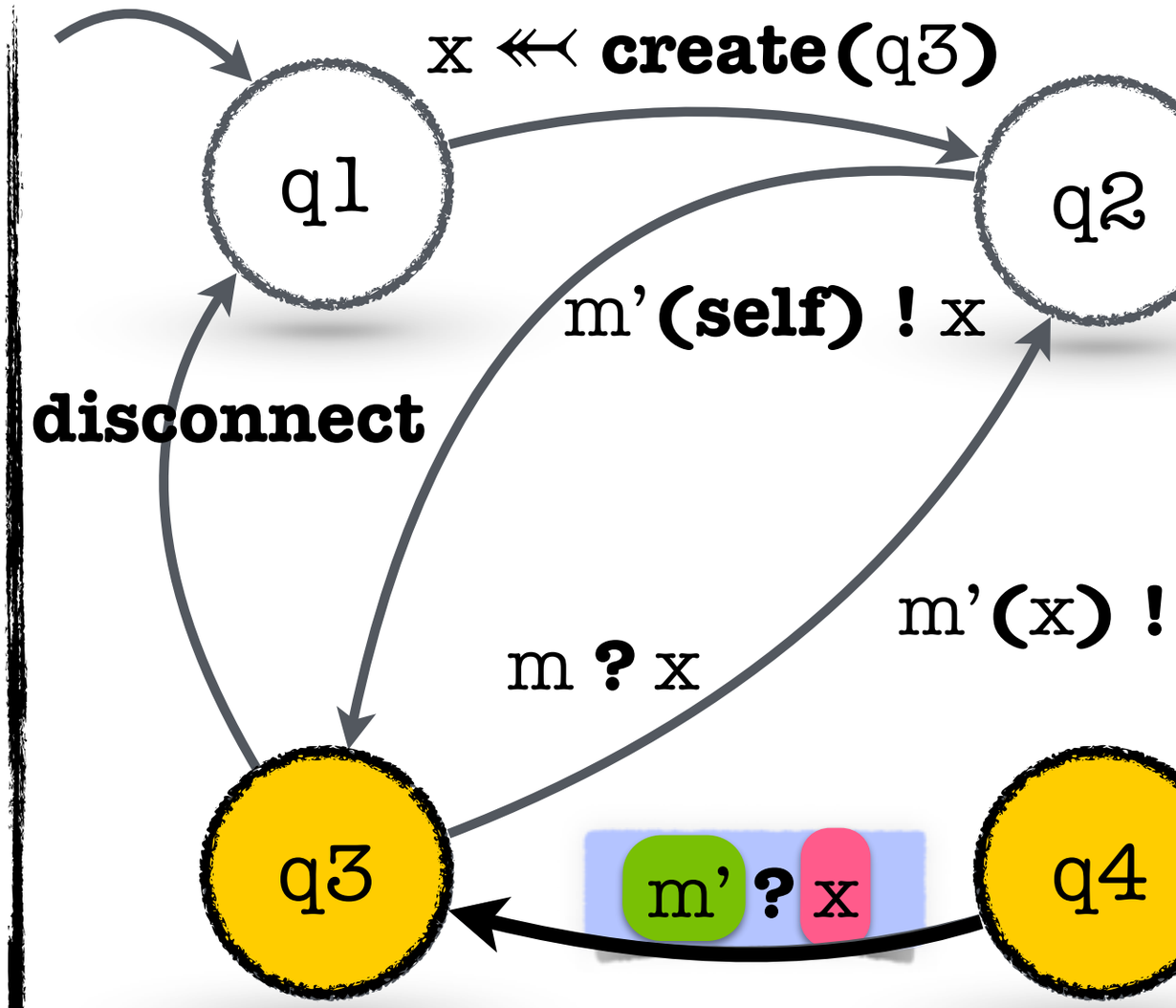
Operational Semantics

Verification of Buffered Dynamic Register Automata

Message Sending



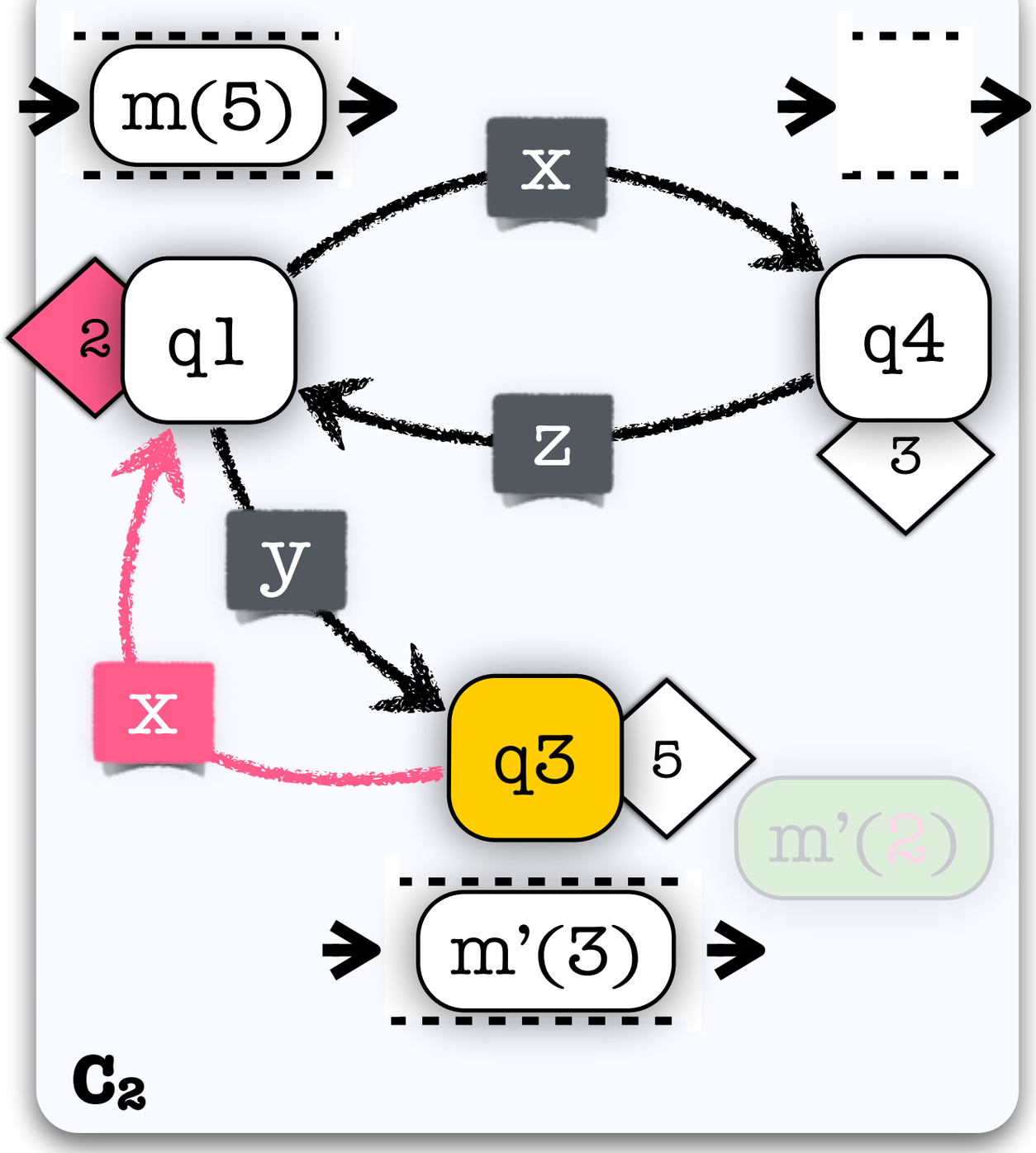
Message Receiving



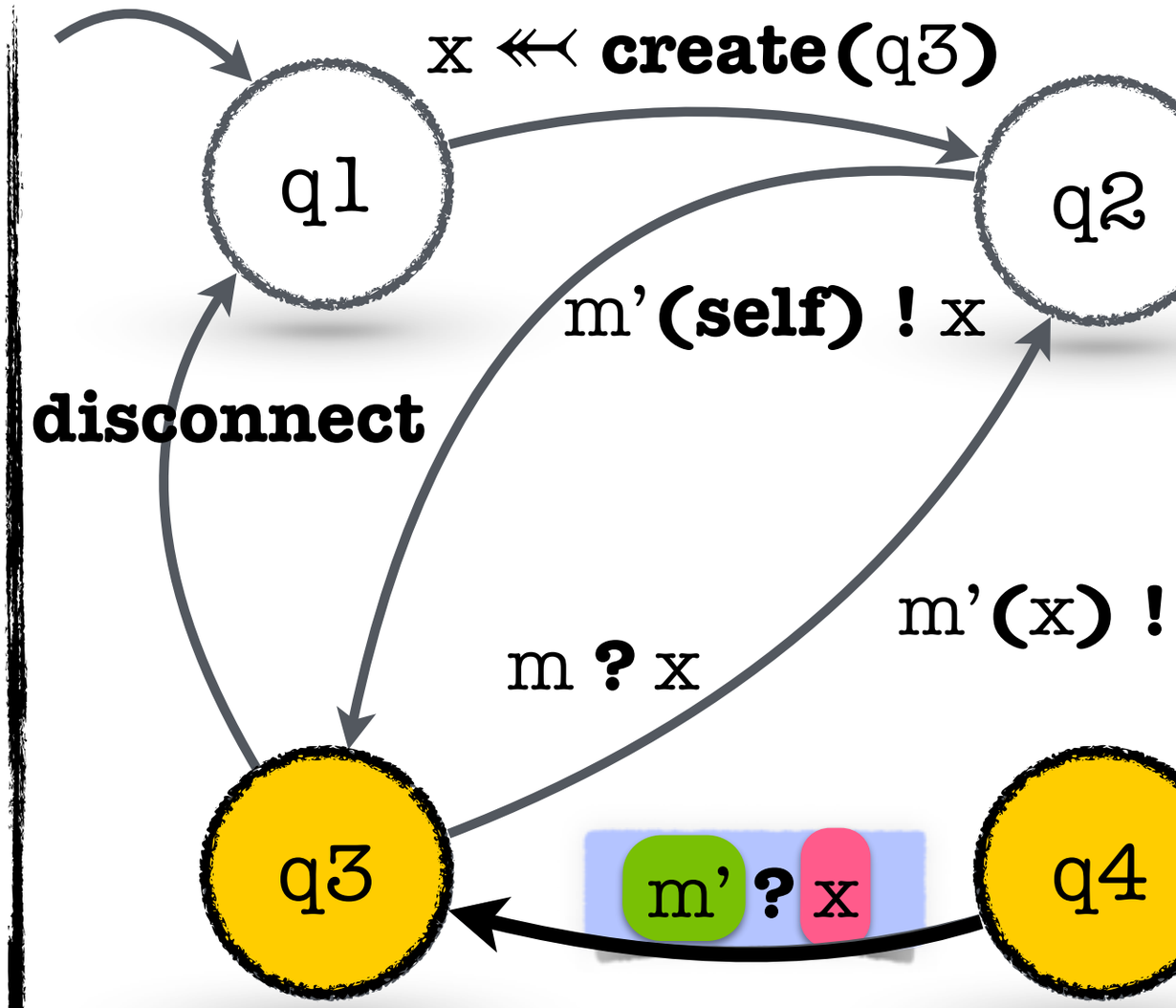
Operational Semantics

Verification of Buffered Dynamic Register Automata

Message Sending



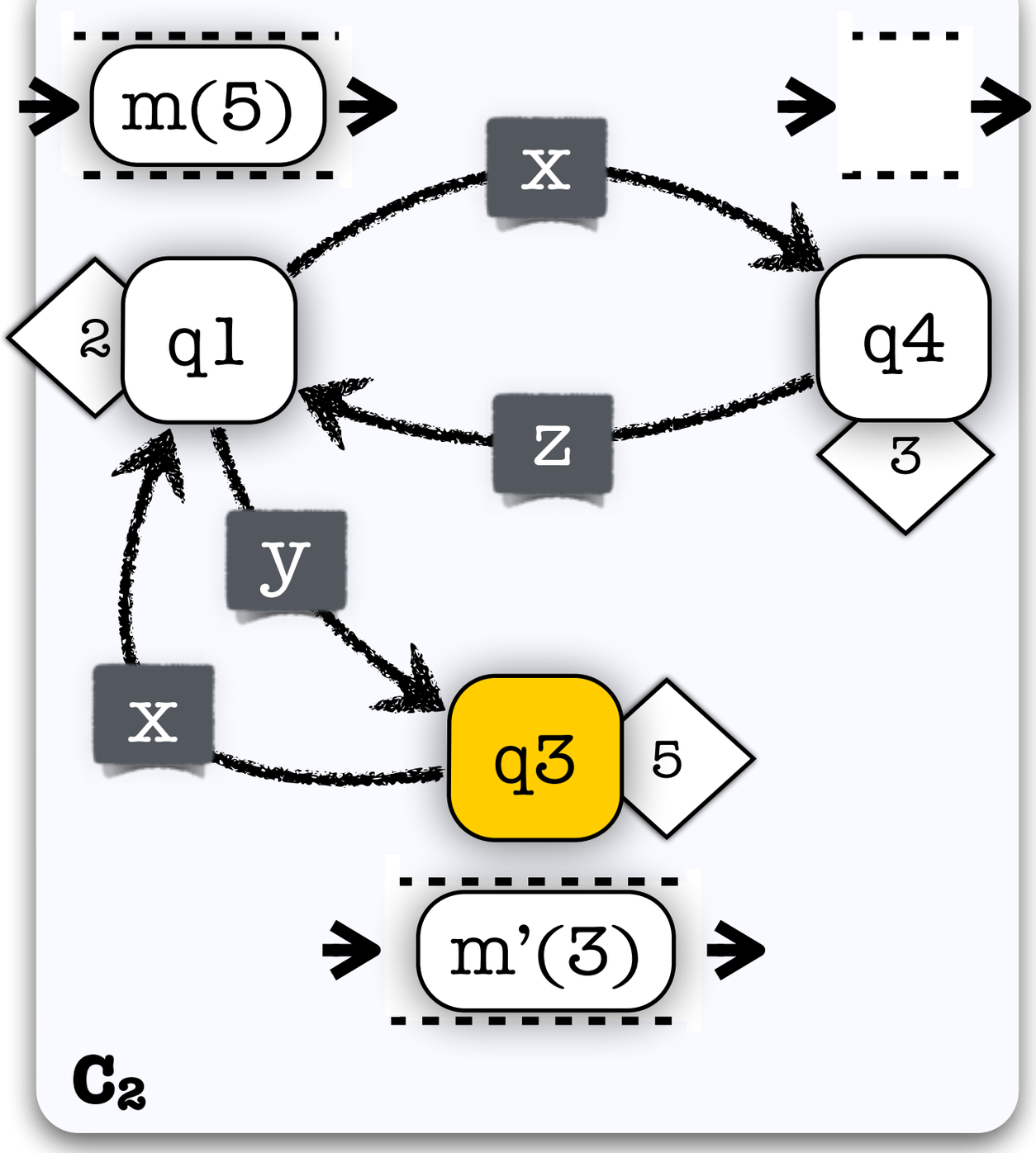
Message Receiving



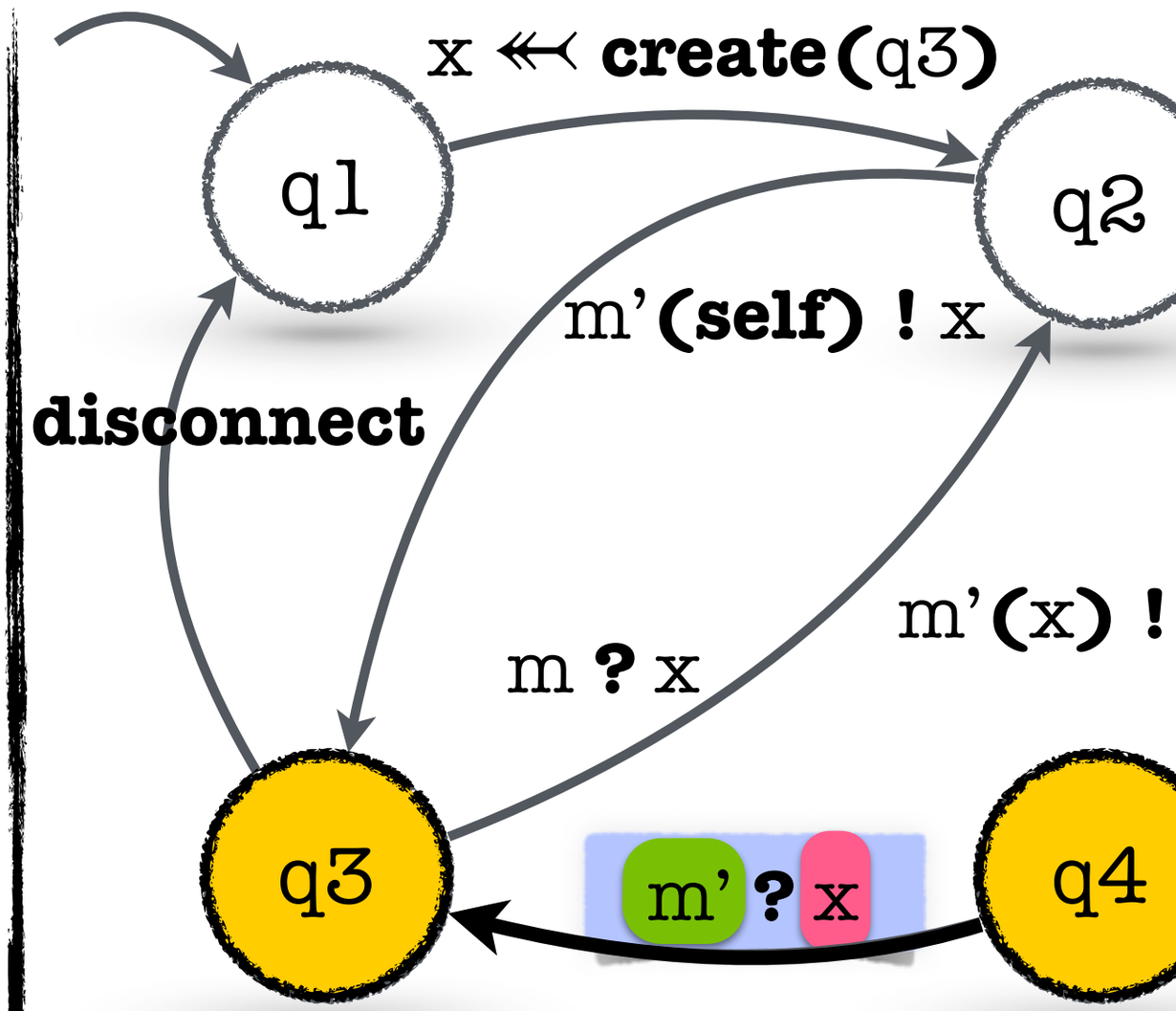
Operational Semantics

Verification of Buffered Dynamic Register Automata

Message Sending

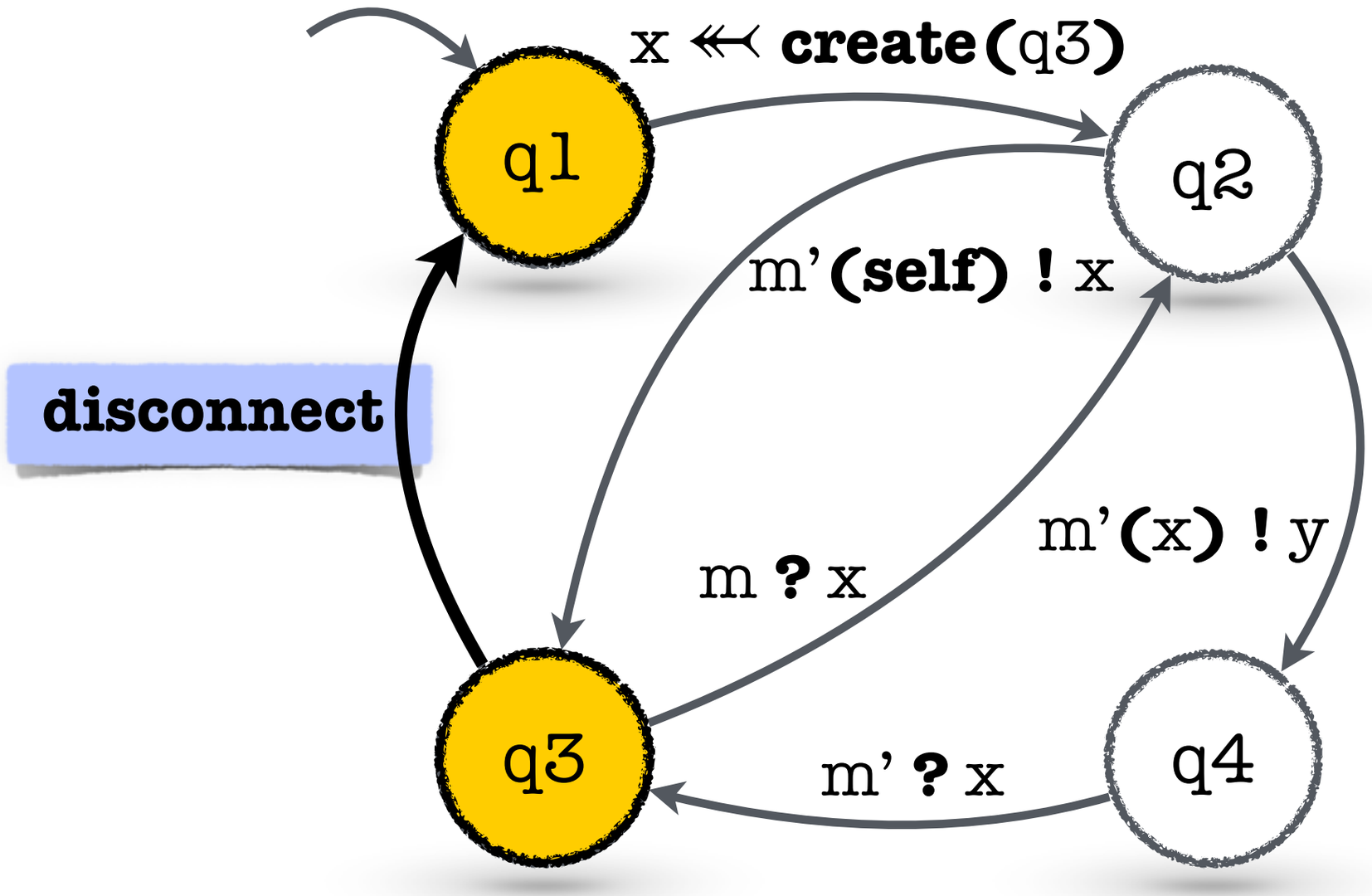


Message Receiving



Operational Semantics

Process Creation **Message Receiving** **Message Sending** **Process Disconnection**

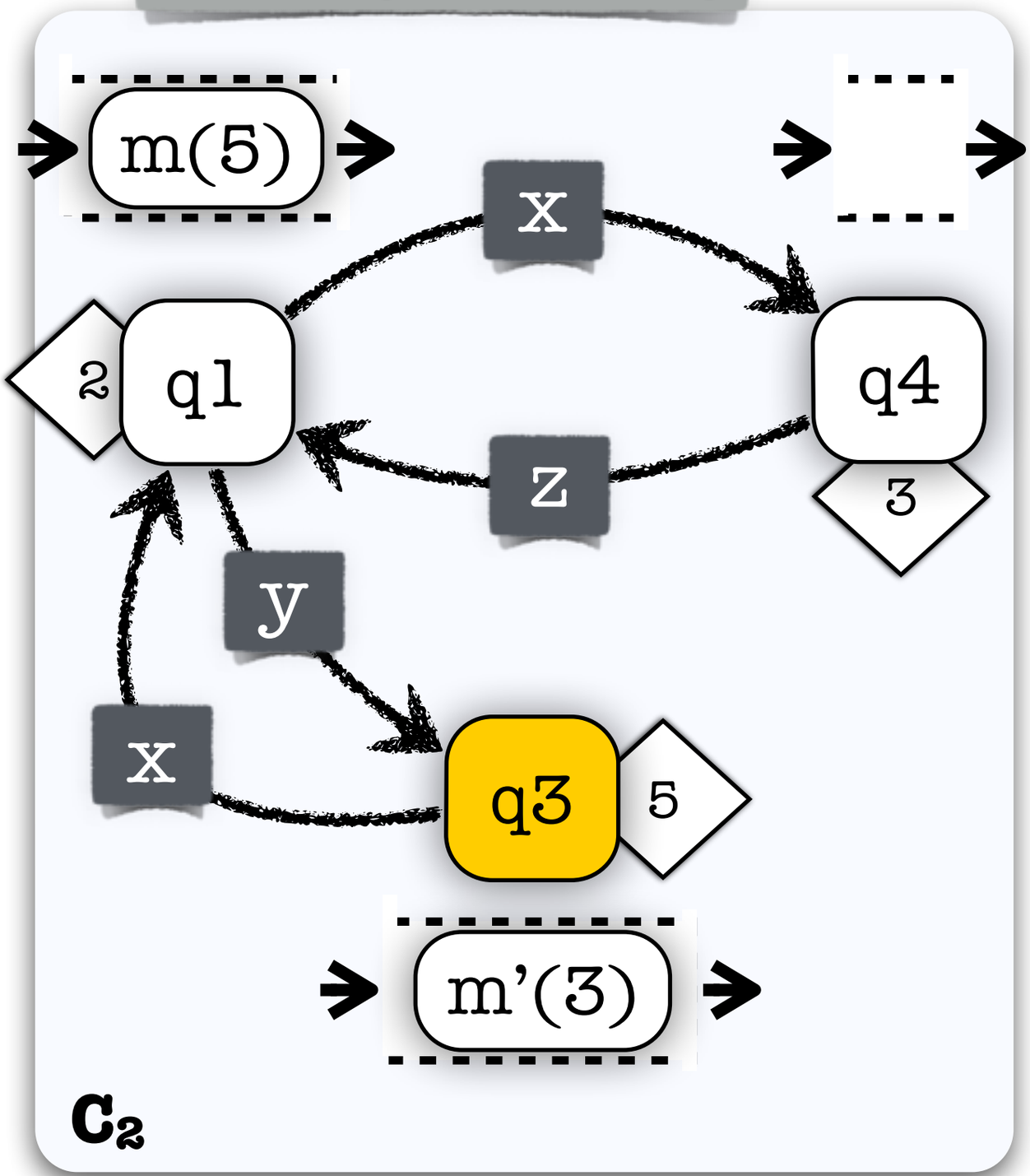


Operational Semantics

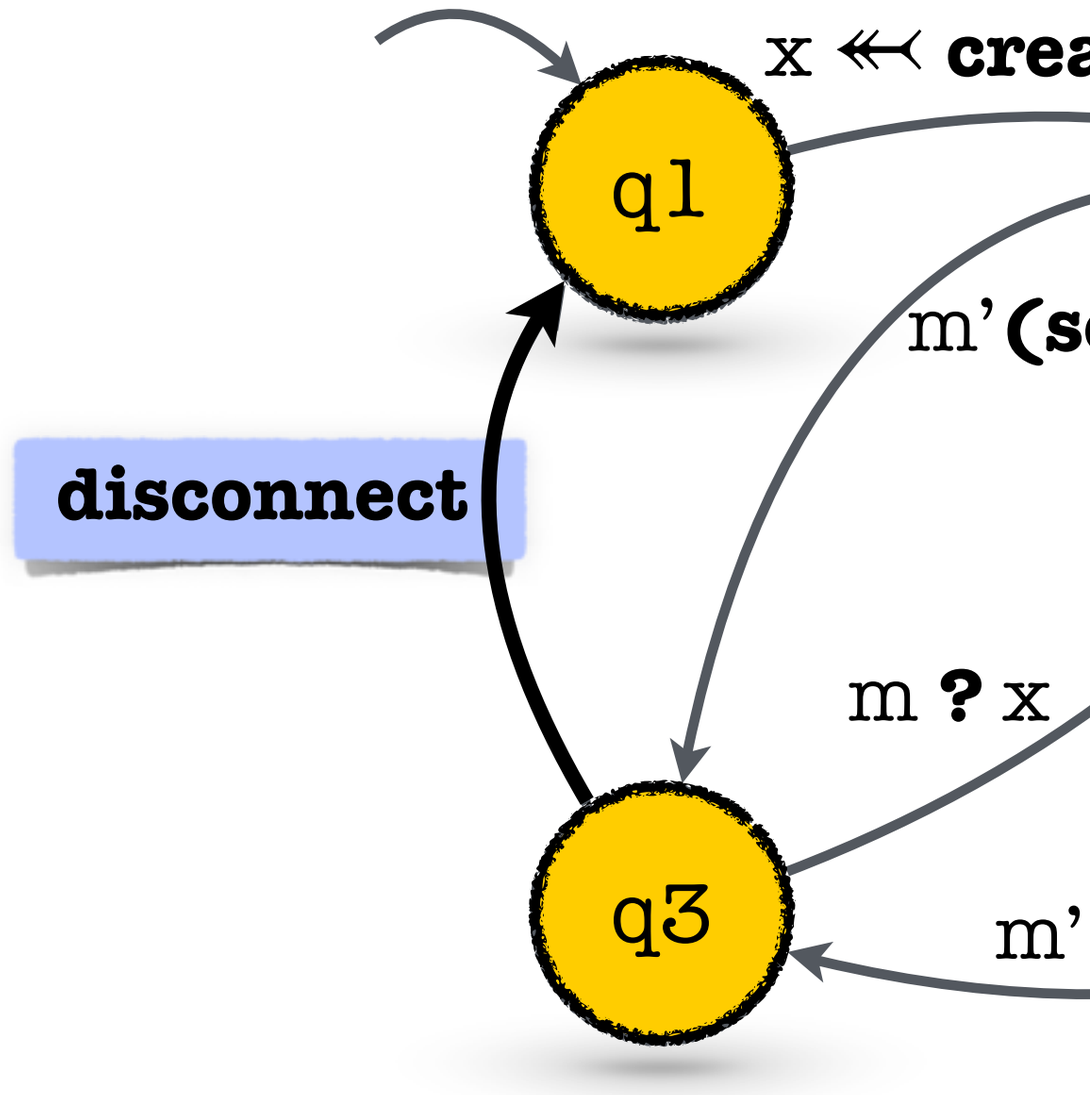
Verification of Buffered Dynamic Register Automata

Message Receiving

Message Sending



Process Disconnection

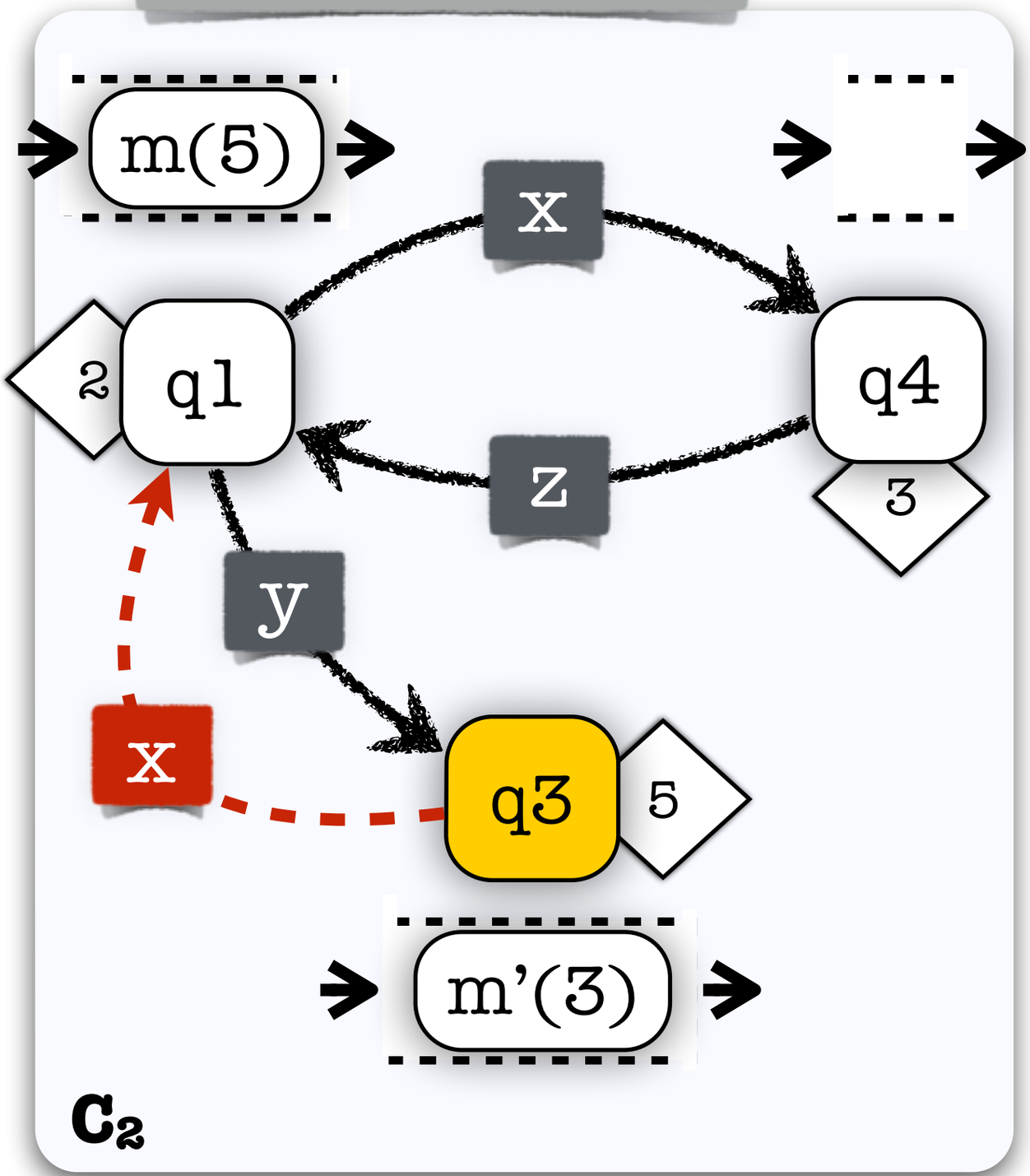


Operational Semantics

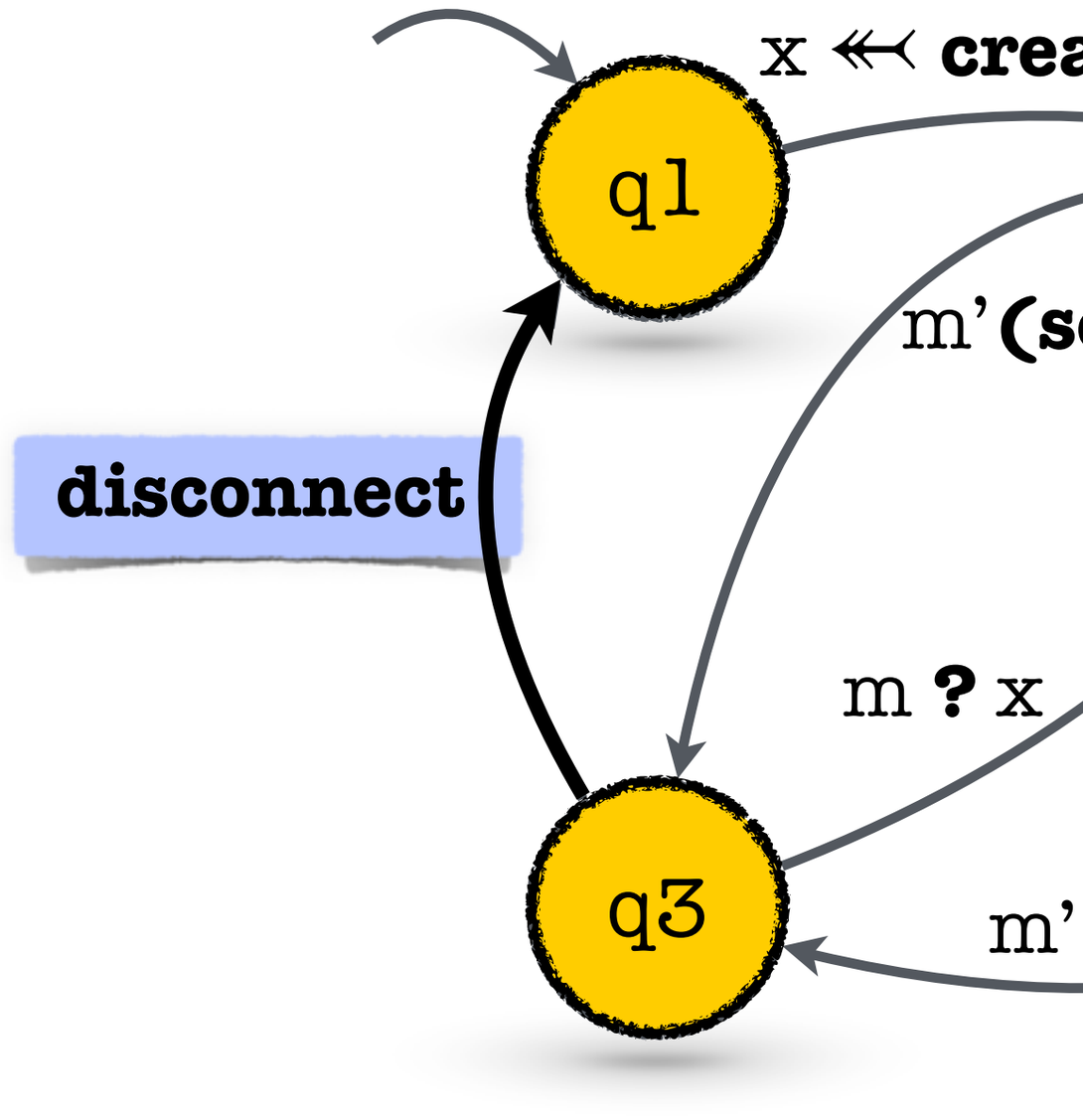
Verification of Buffered Dynamic Register Automata

Message Receiving

Message Sending



Process Disconnection

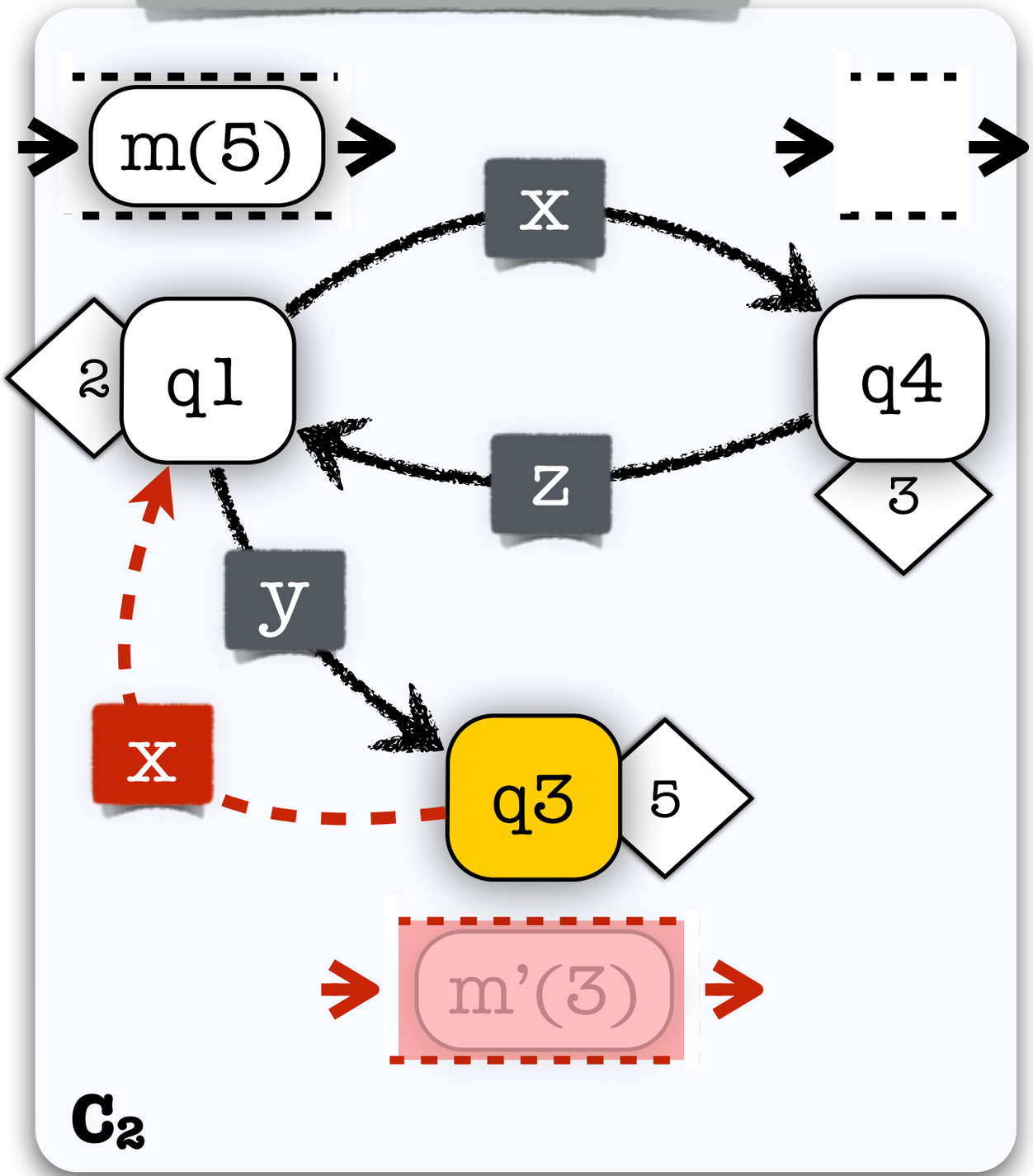


Operational Semantics

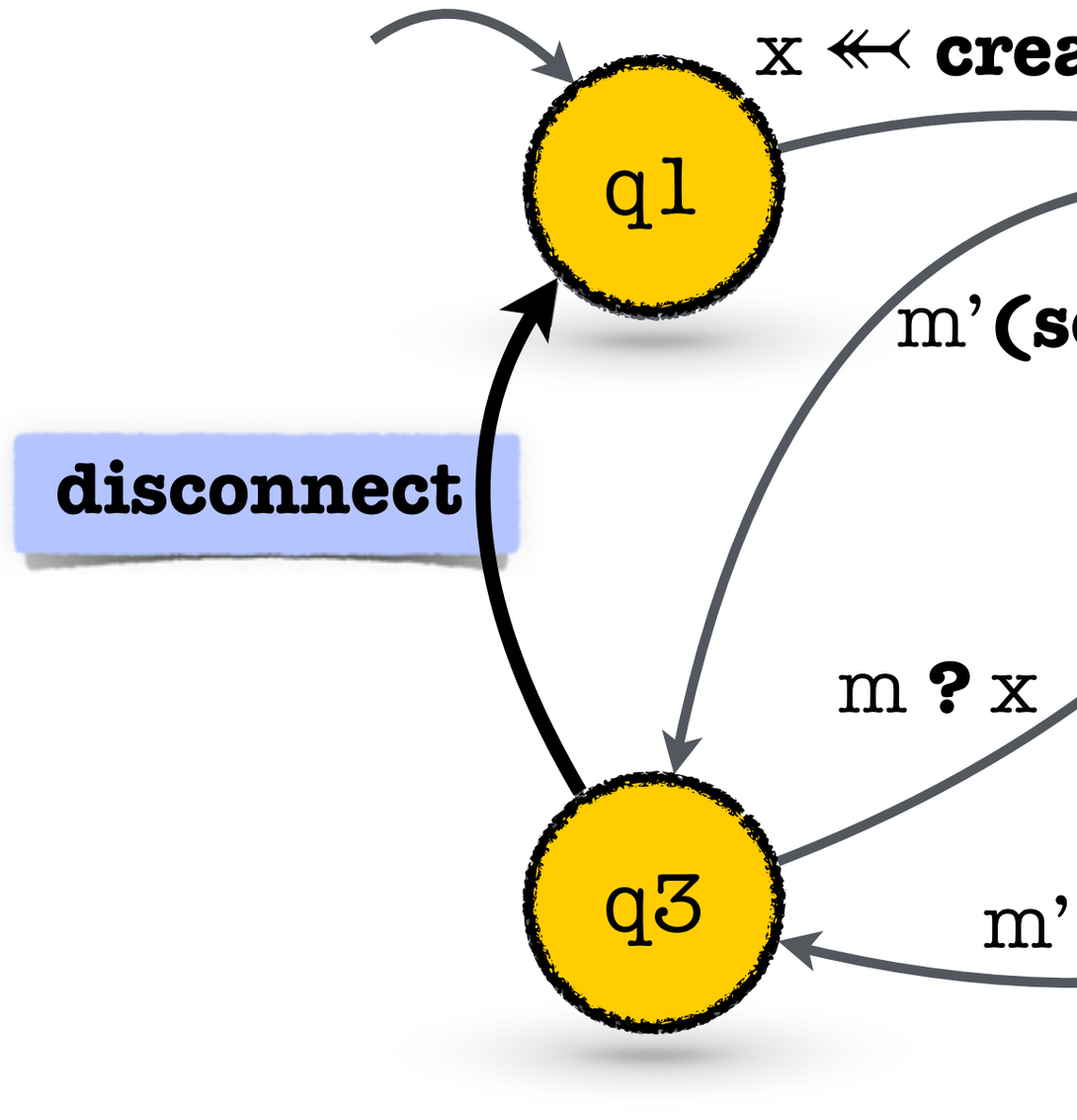
Verification of Buffered Dynamic Register Automata

Message Receiving

Message Sending



Process Disconnection



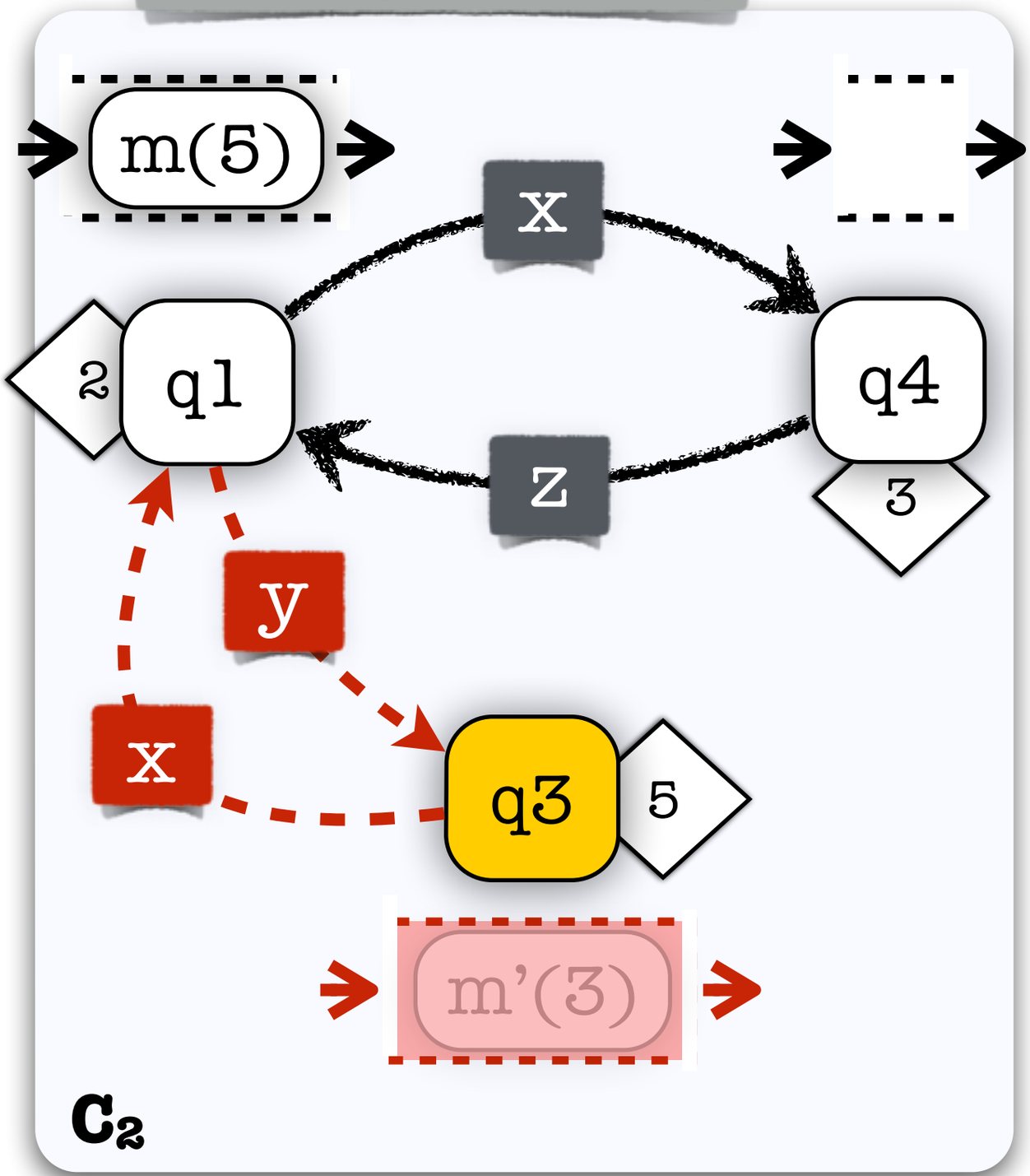
Operational Semantics

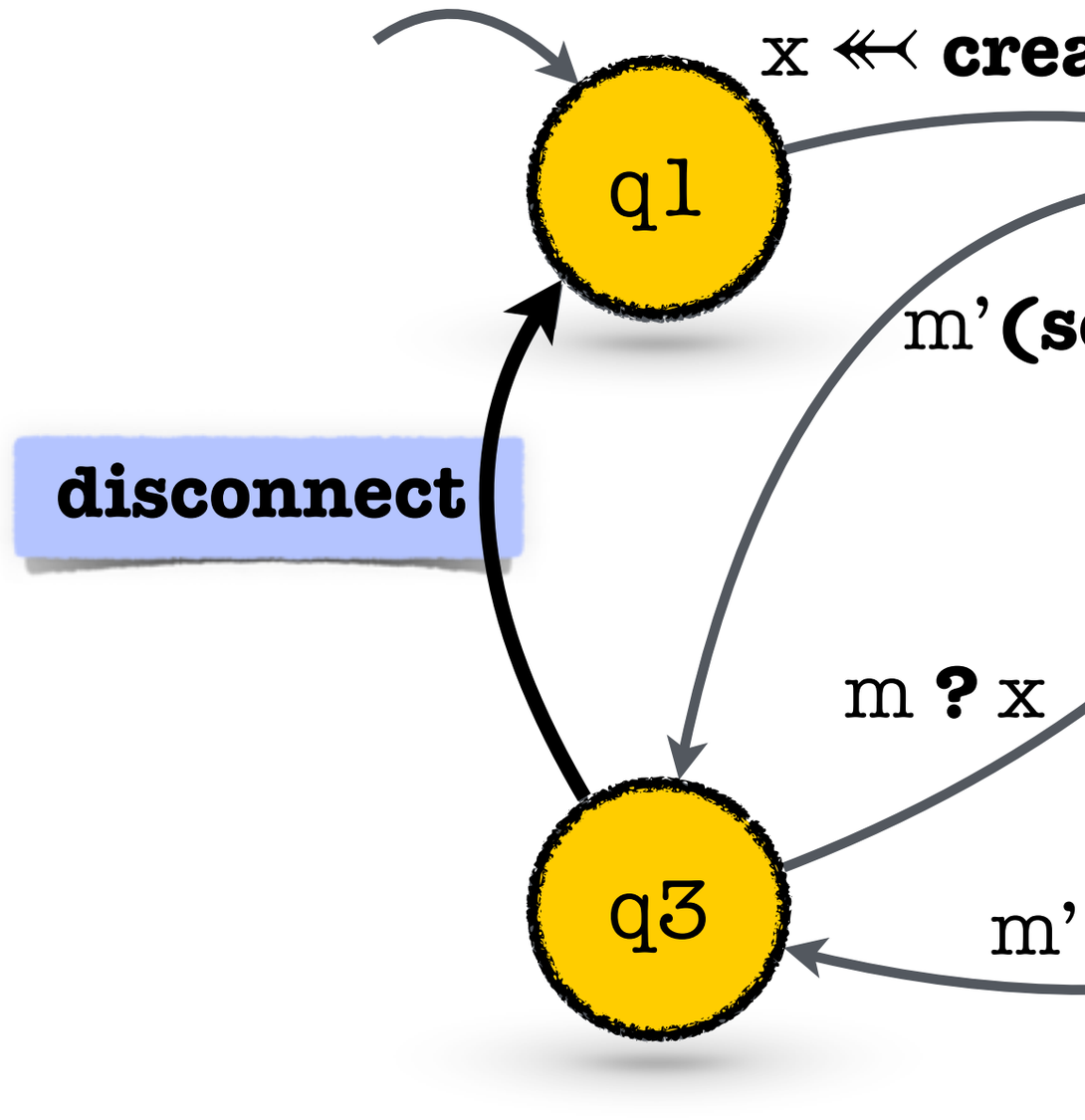
Verification of Buffered Dynamic Register Automata

Message Sending

Message Receiving

Process Disconnection



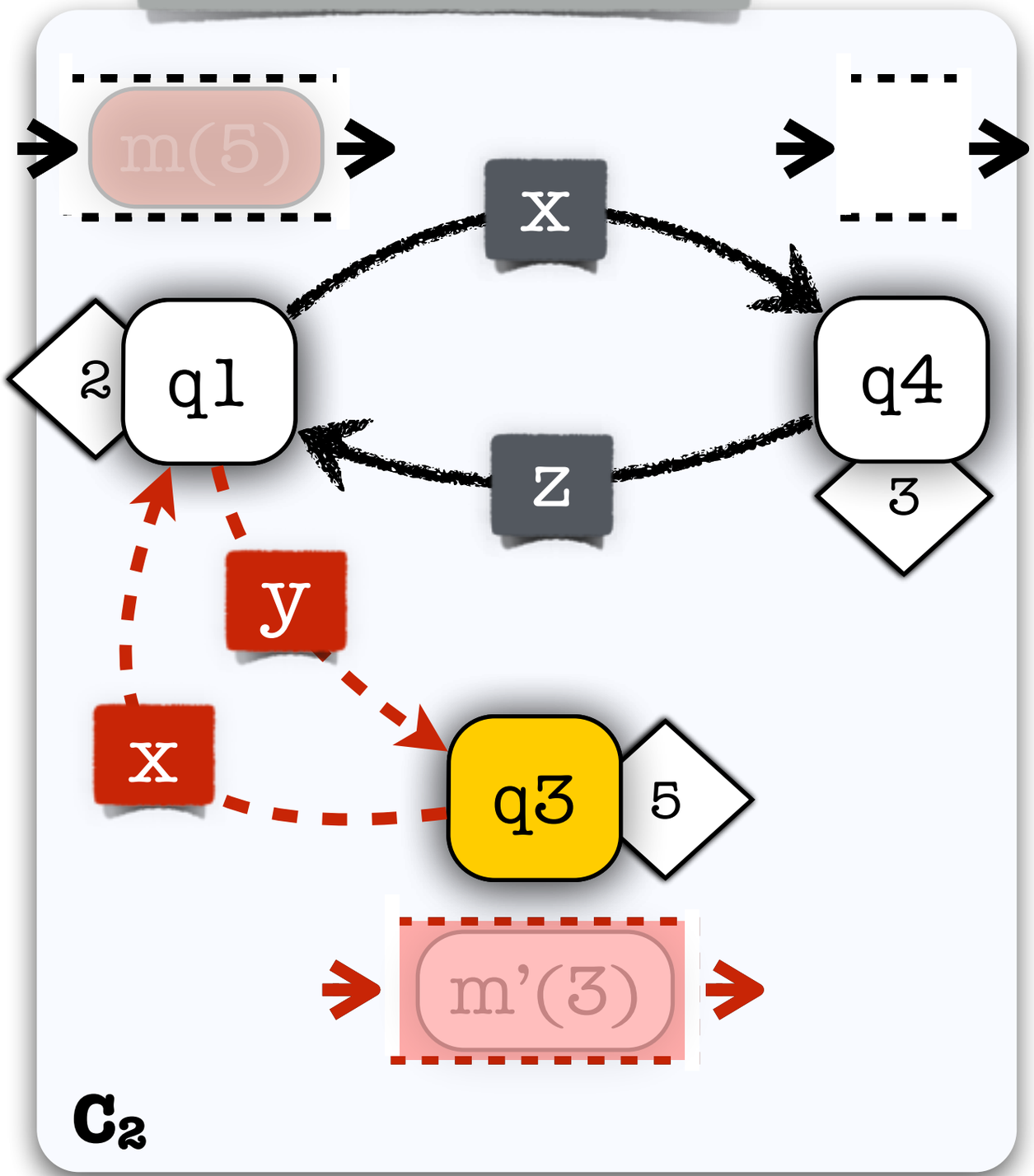


Operational Semantics

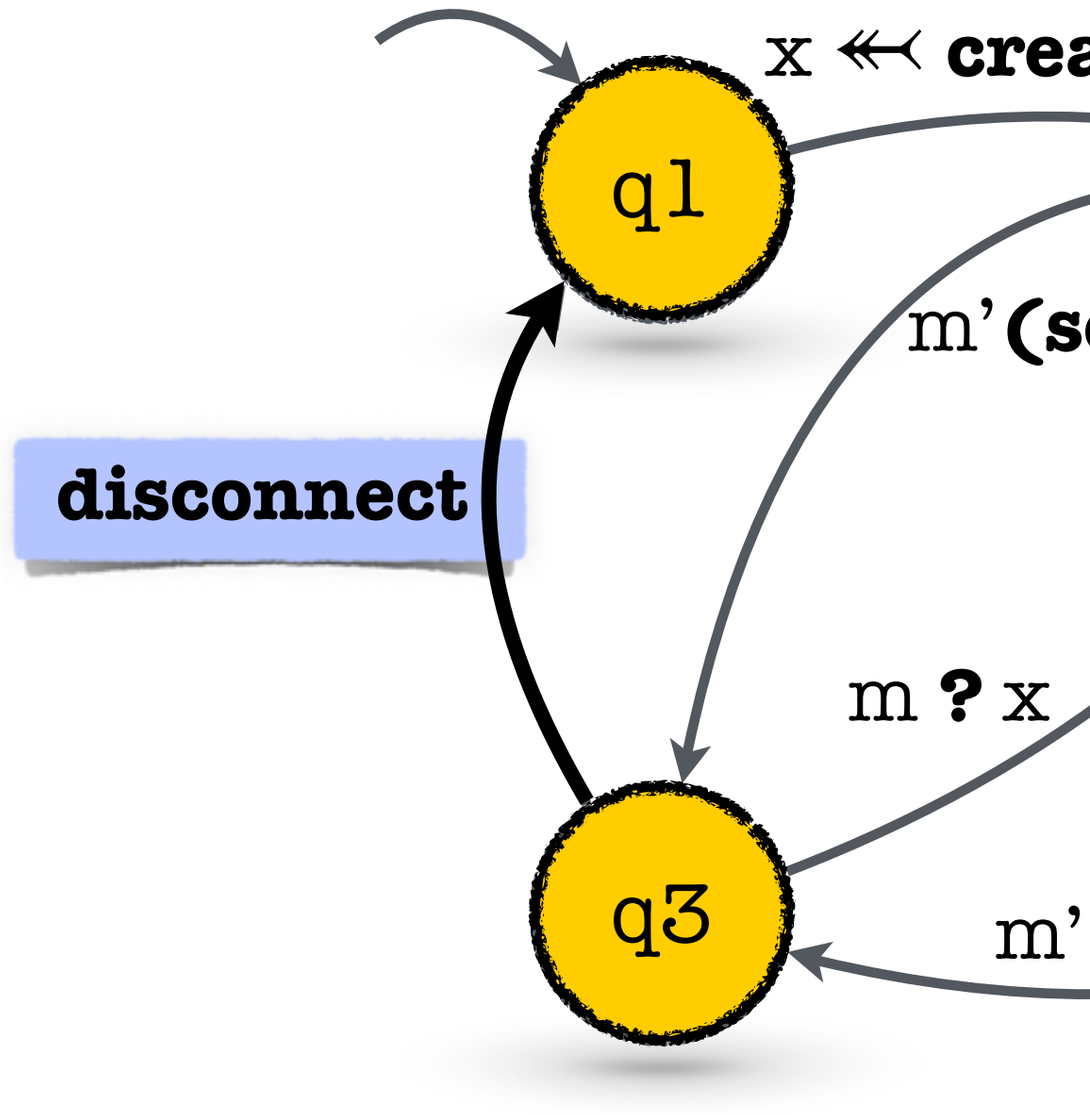
Verification of Buffered Dynamic Register Automata

Message Receiving

Message Sending



Process Disconnection

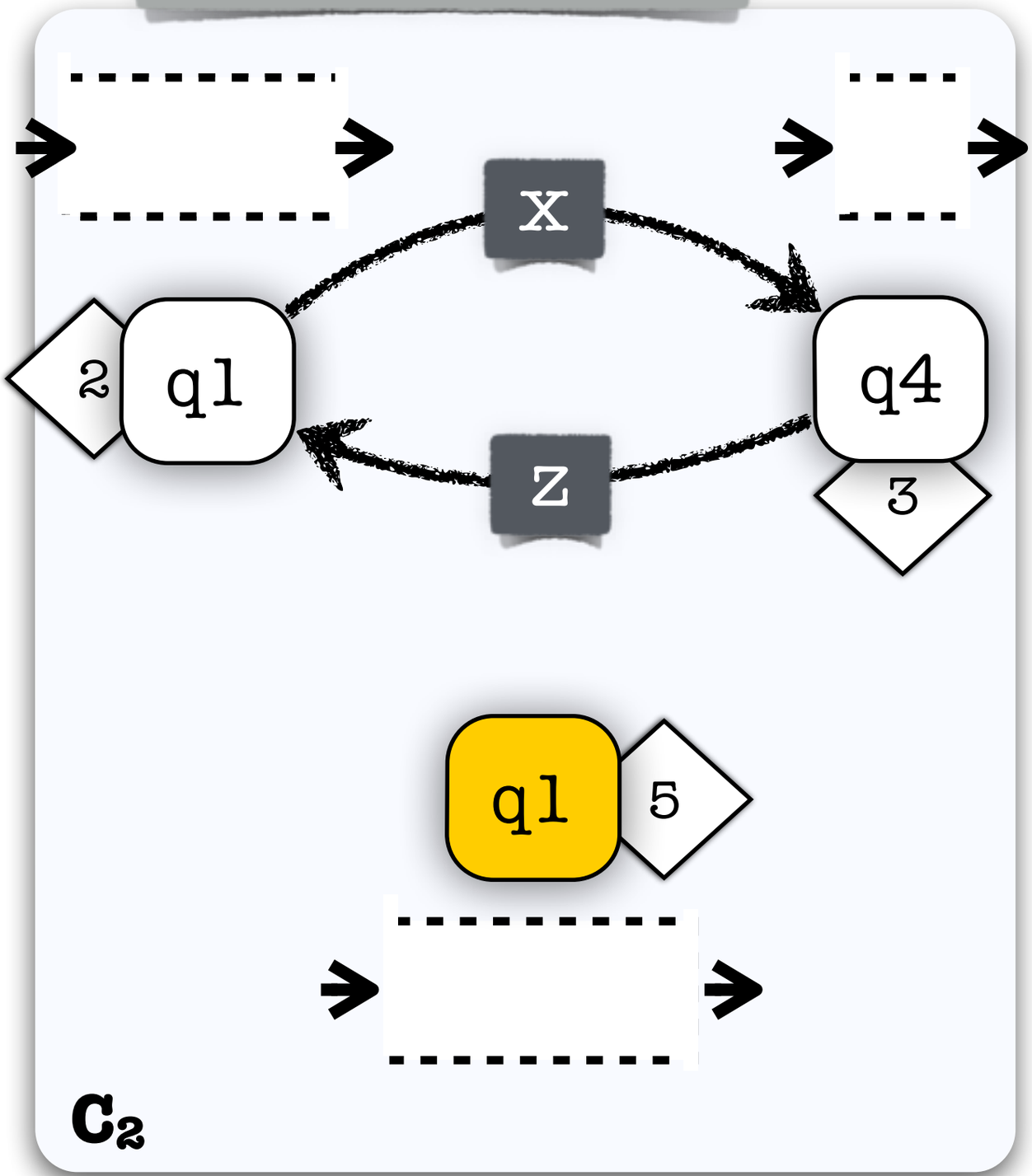


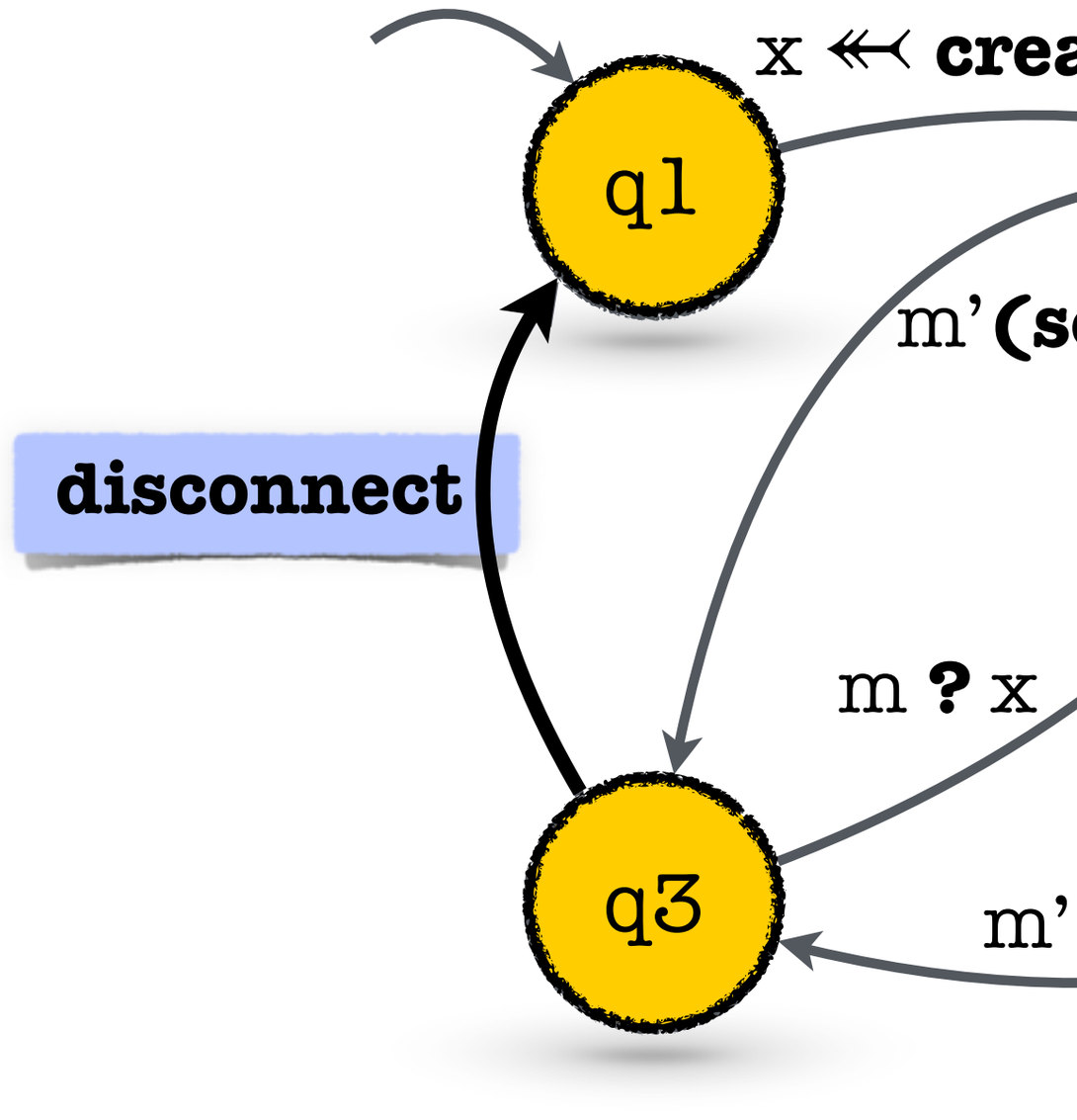
Operational Semantics

Message Receiving

Message Sending

Process Disconnection

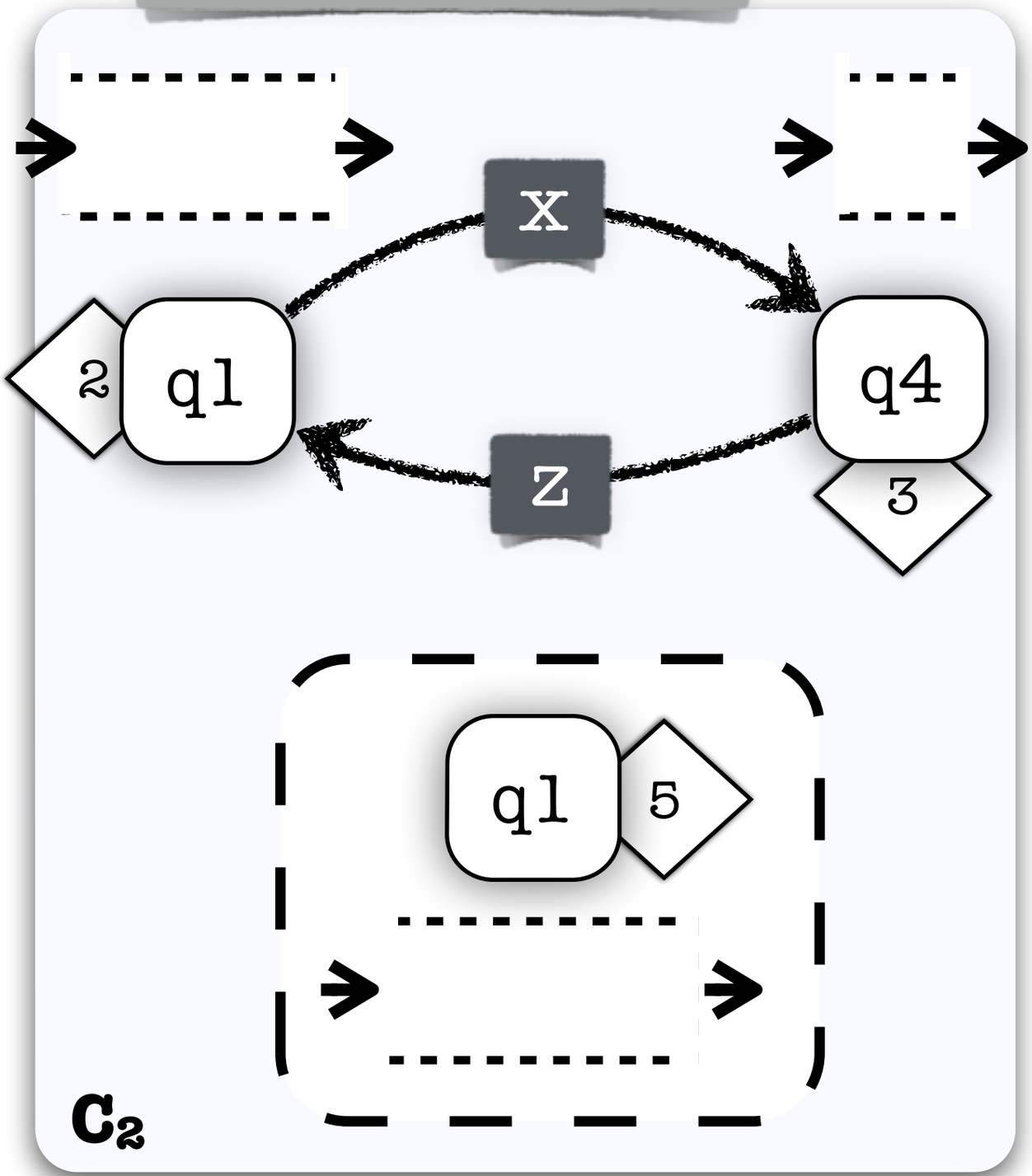




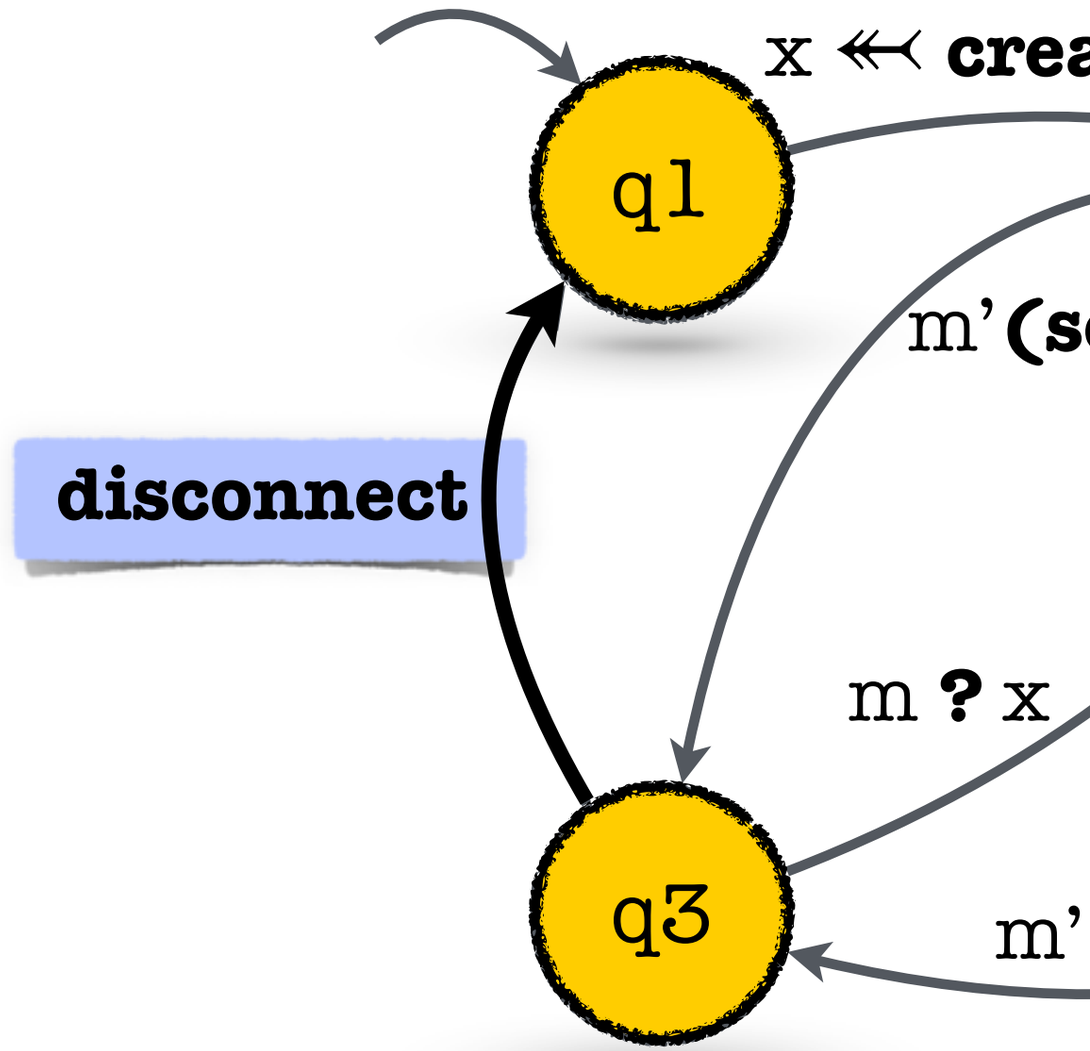
Operational Semantics

Pro… Message Receiving

Message Sending



Process Disconnection

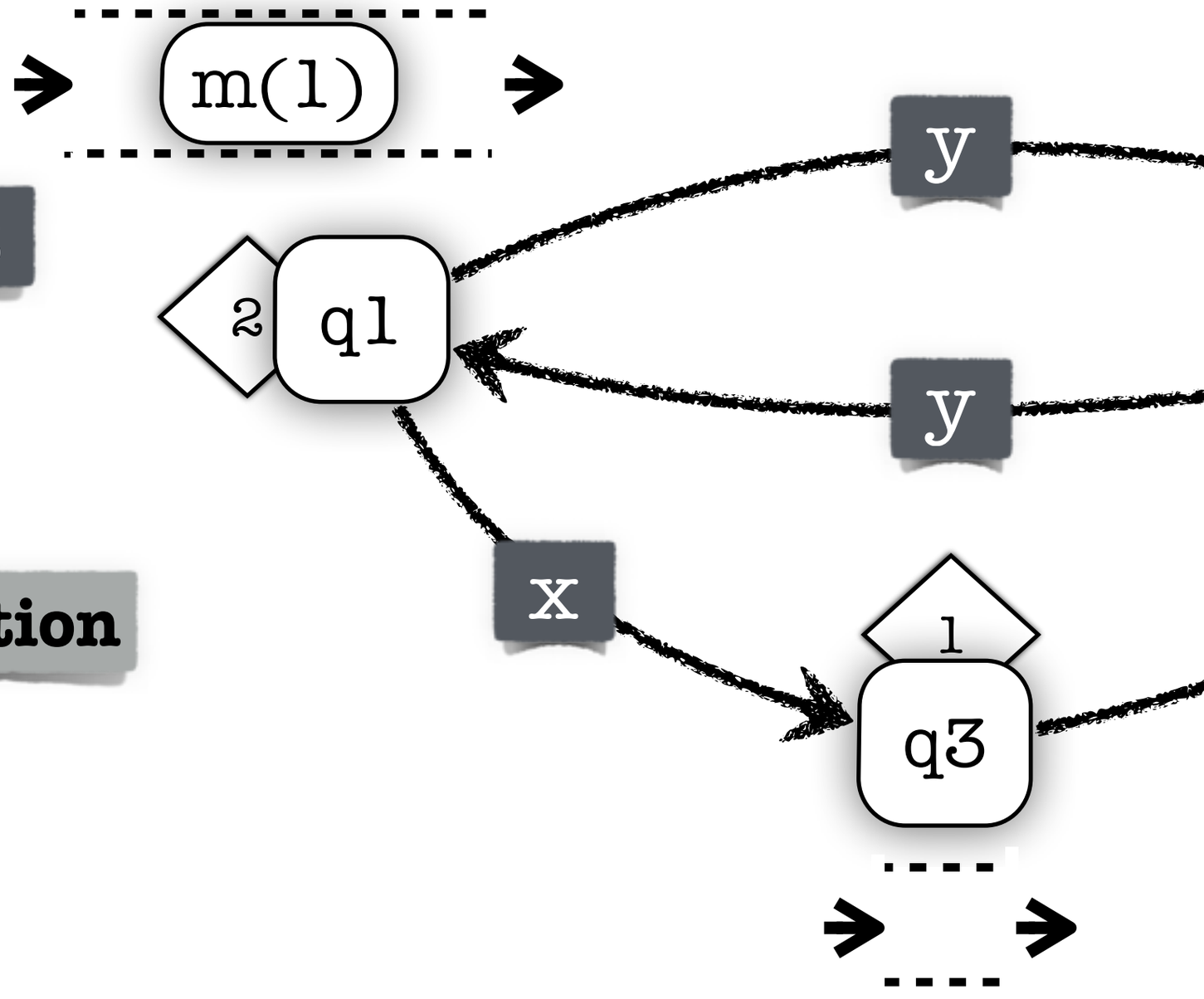


Buffered Dynamic Register Automata

Formal Model

Operational Semantics

- **Process Creation**
- **Message Sending**
- **Message Receiving**
- **Process Disconnection**



Buffered Dynamic Register Automata

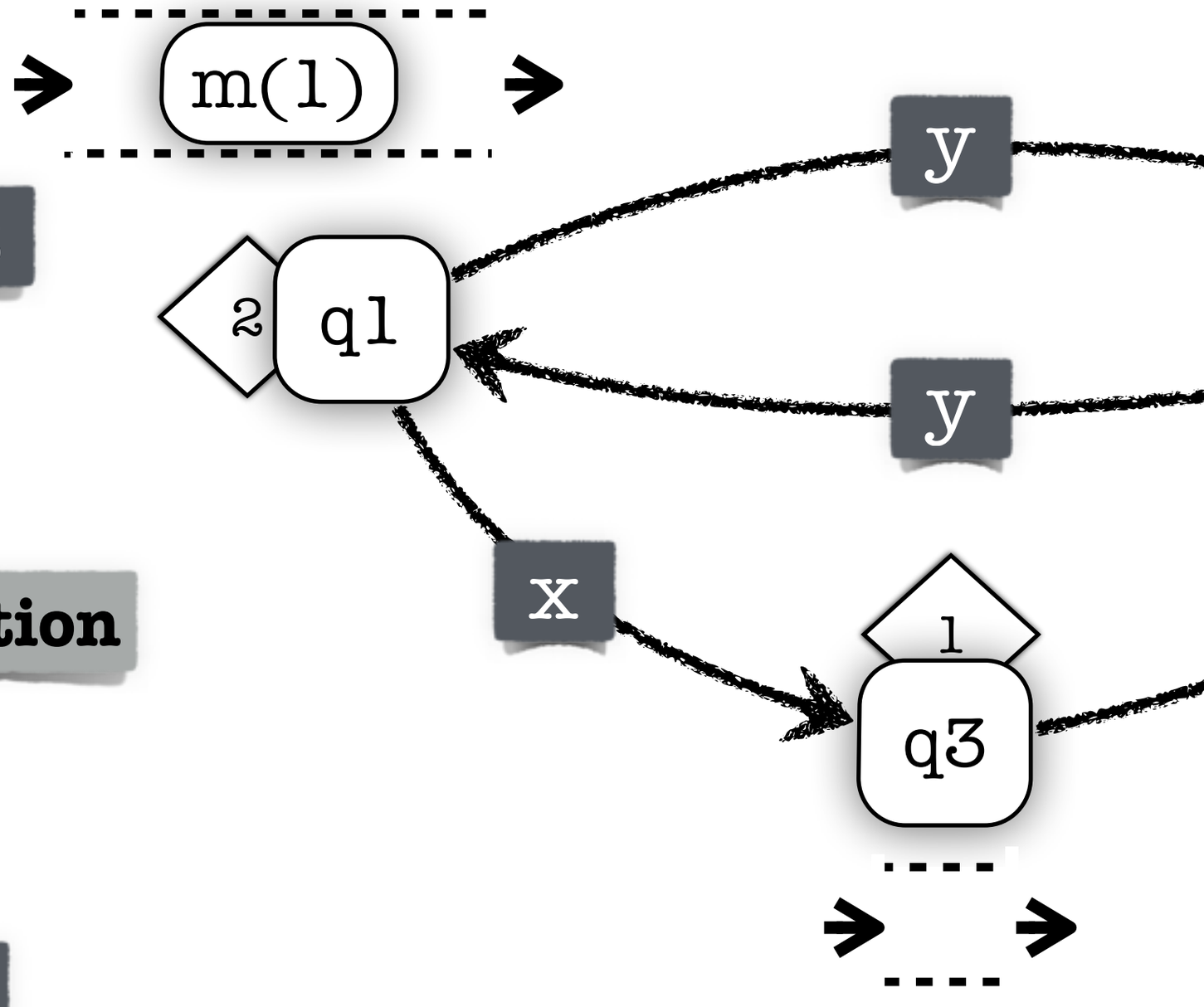
Formal Model

Operational Semantics

- **Process Creation**
- **Message Sending**
- **Message Receiving**
- **Process Disconnection**

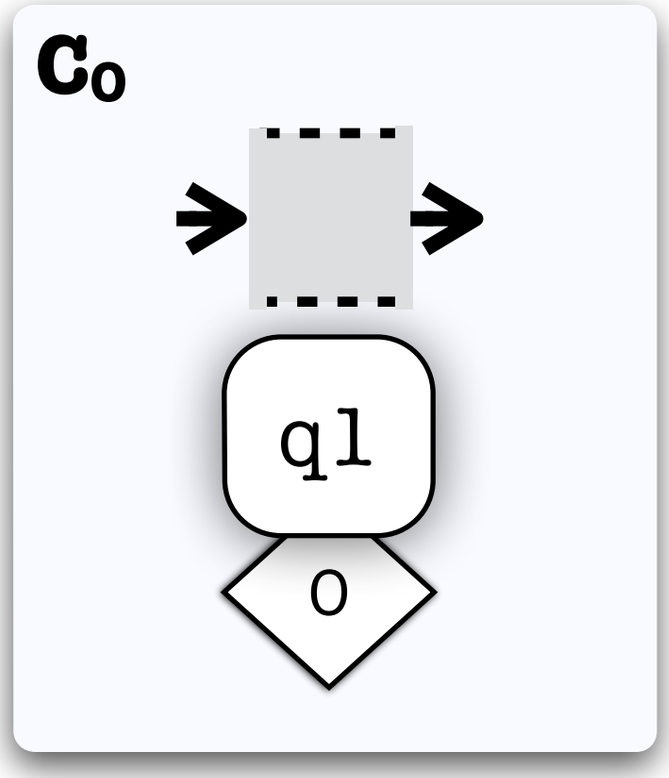
Problem:

State Reachability



State Reachability

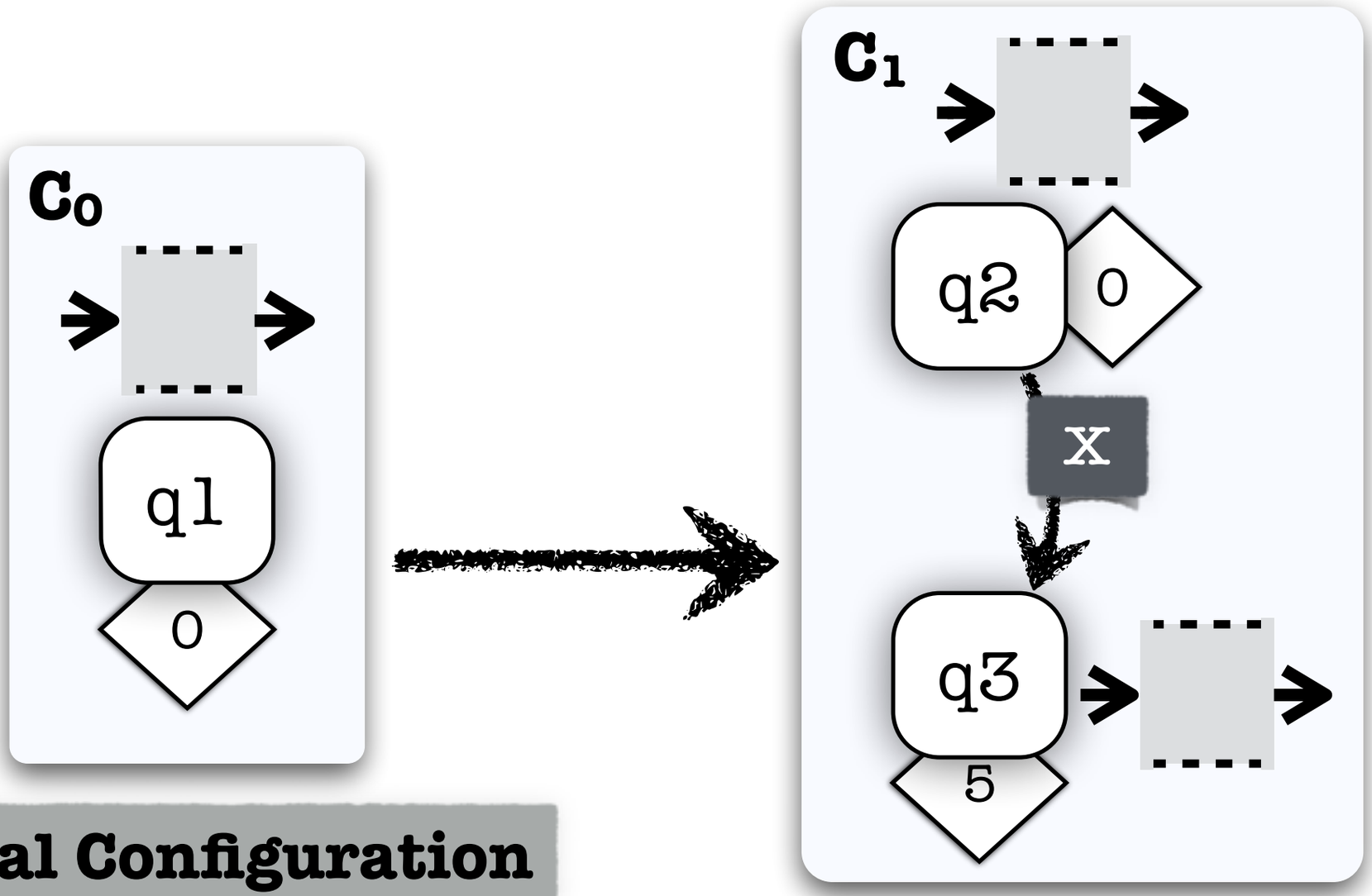
Verification of Buffered Dynamic Register Automata



Initial Configuration

State Reachability

Verification of Buffered Dynamic Register Automata

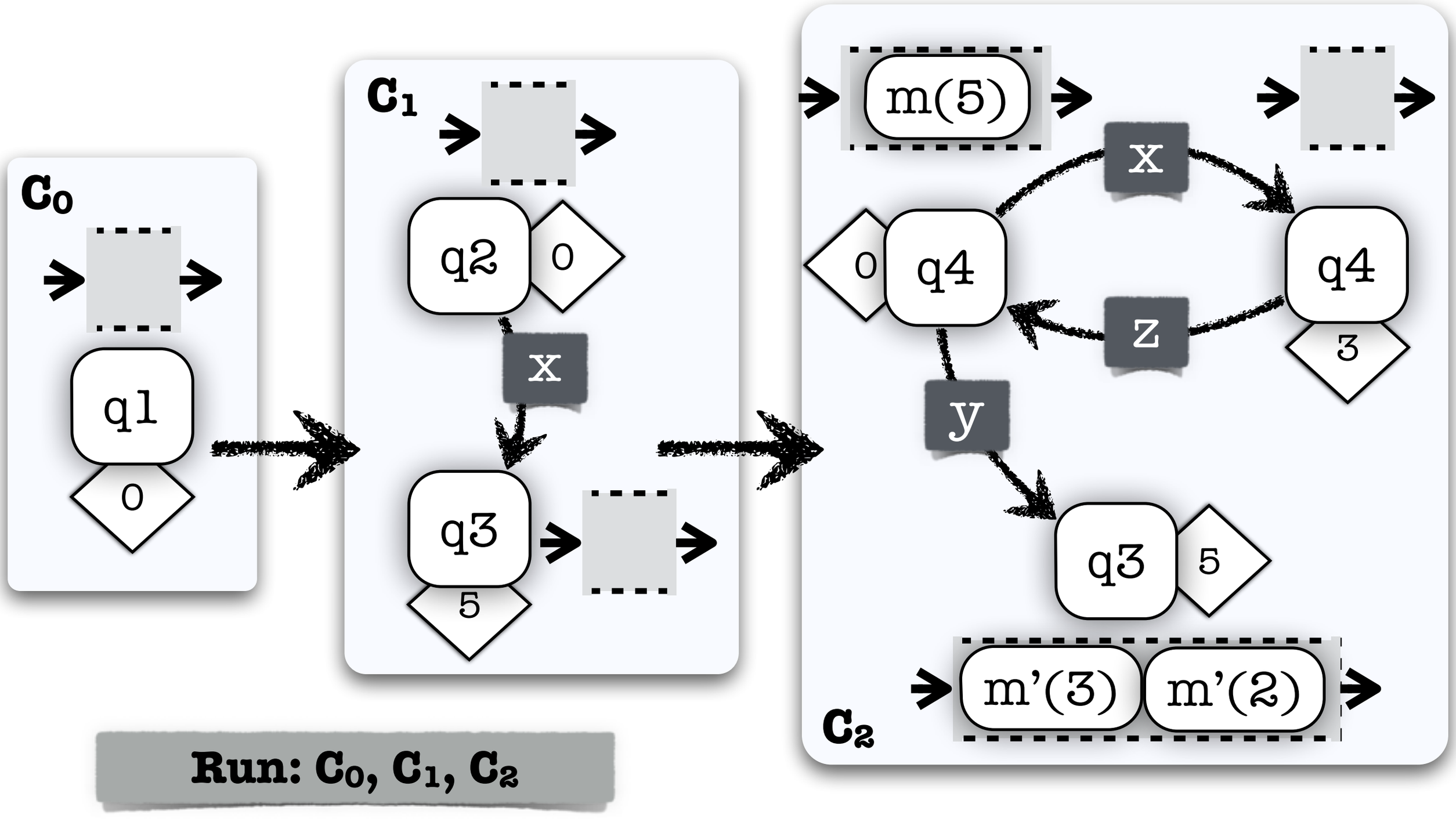


Initial Configuration

Run: C_0, C_1

State Reachability

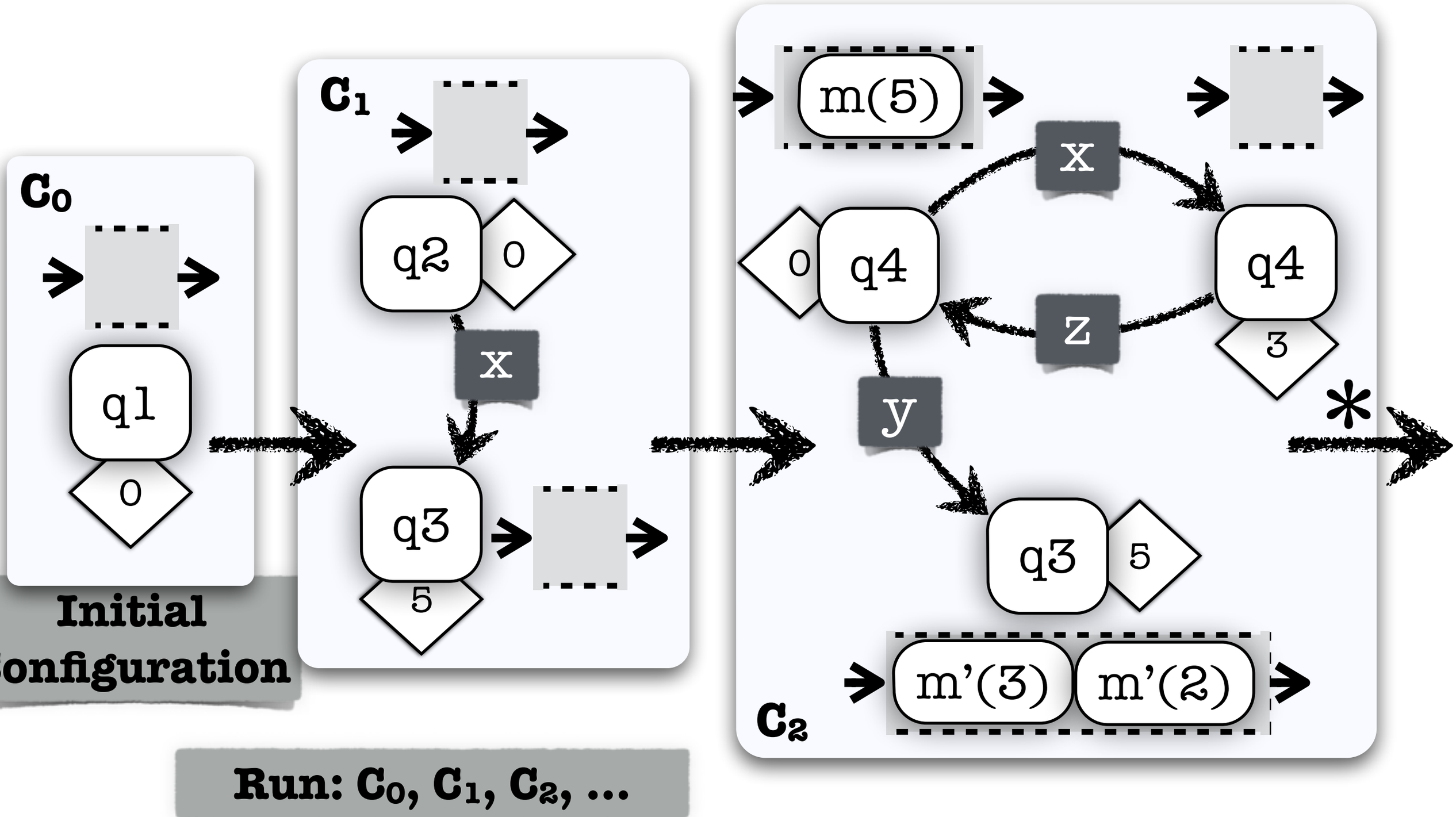
Verification of Buffered Dynamic Register Automata



Run: C_0, C_1, C_2

State Reachability

Verification of Buffered Dynamic Register Automata



State Reachability

Initial Configuration **Run: $C_0, C_1, C_2, \ldots, C_n$**

State Reachability Problem:

State Reachability

State Reachability Problem:

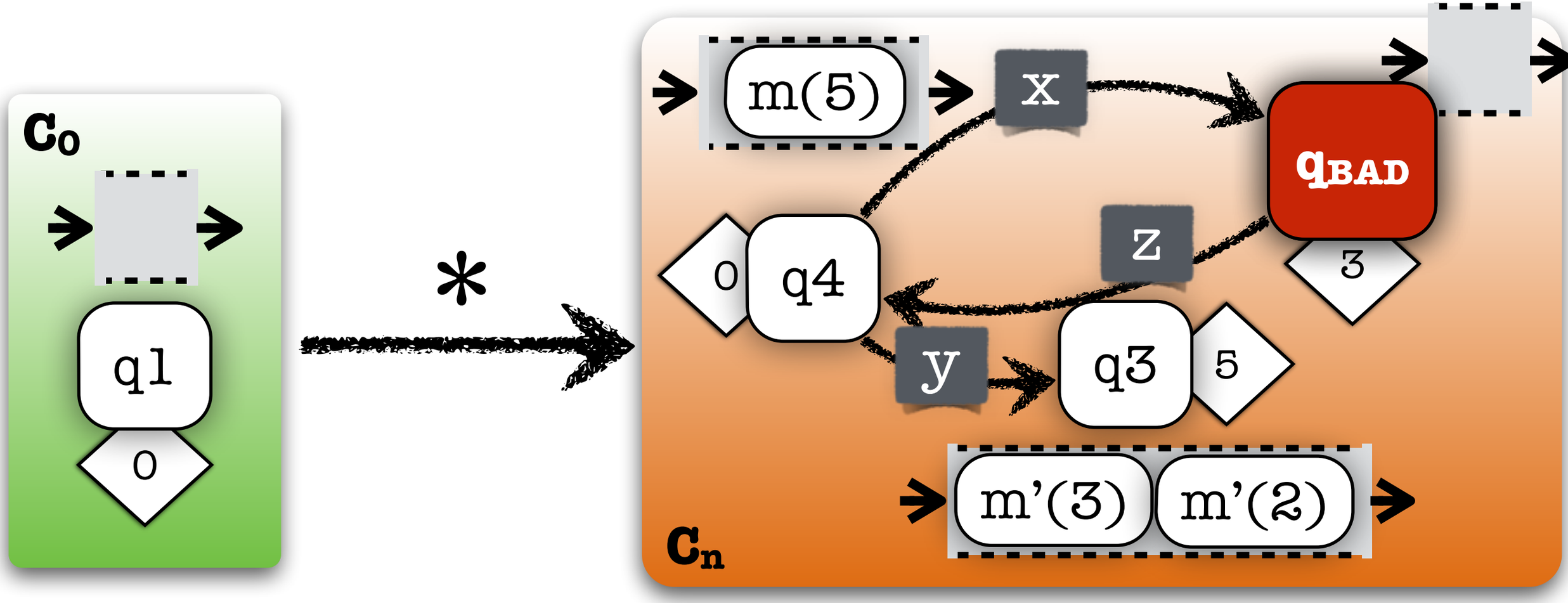
Given q_{BAD}, is there a run starting from an **initial configuration** that reaches **a configuration** where q_{BAD} occurs?

State Reachability

Verification of Buffered Dynamic Register Automata

State Reachability Problem:

Given q_{BAD}, is there a run starting from an **initial configuration** that reaches **a configuration** where q_{BAD} occurs?



State Reachability

Verification of Buffered Dynamic Register Automata

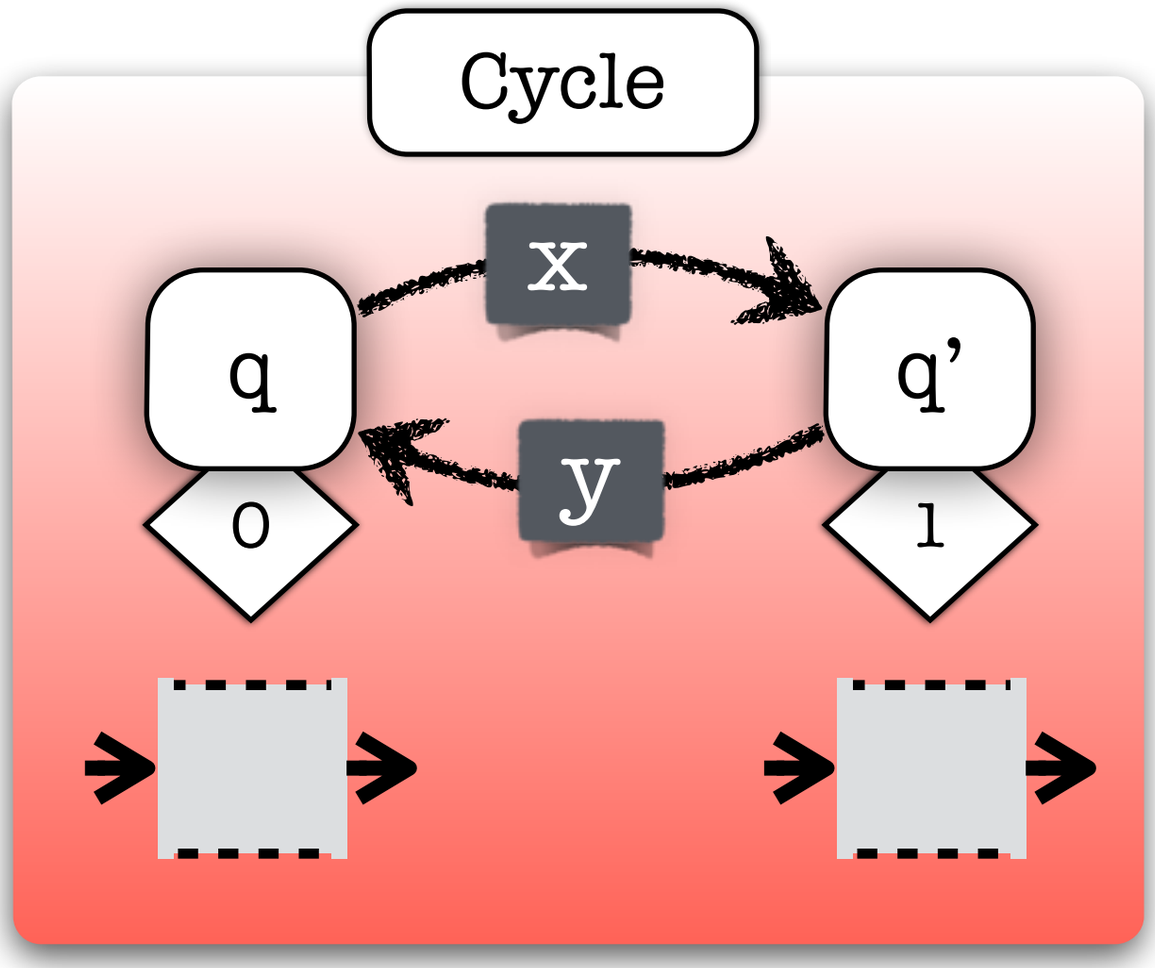
State Reachability Problem:

Undecidable

State Reachability

Verification of Buffered Dynamic Register Automata

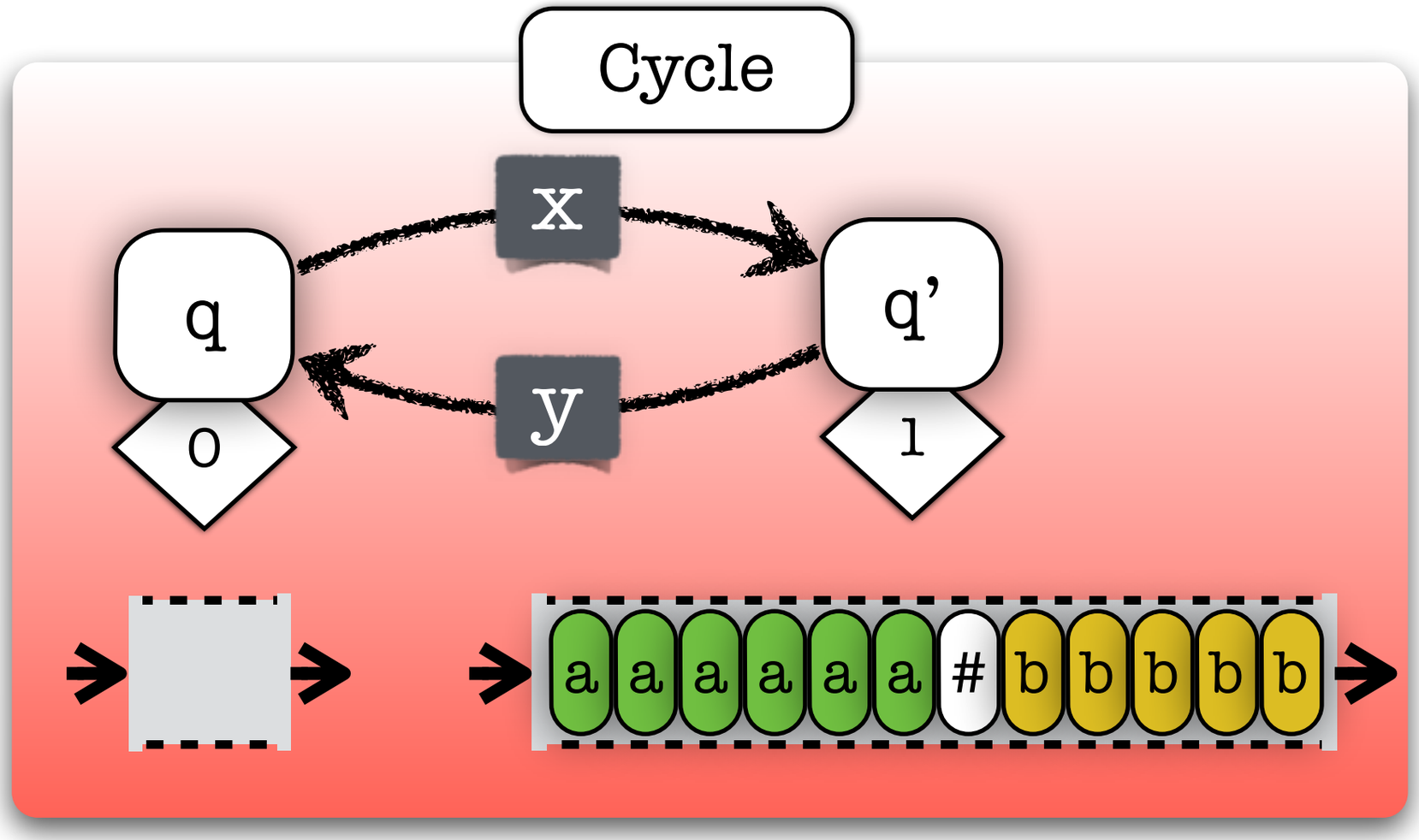
State Reachability Problem: **Undecidable**



State Reachability

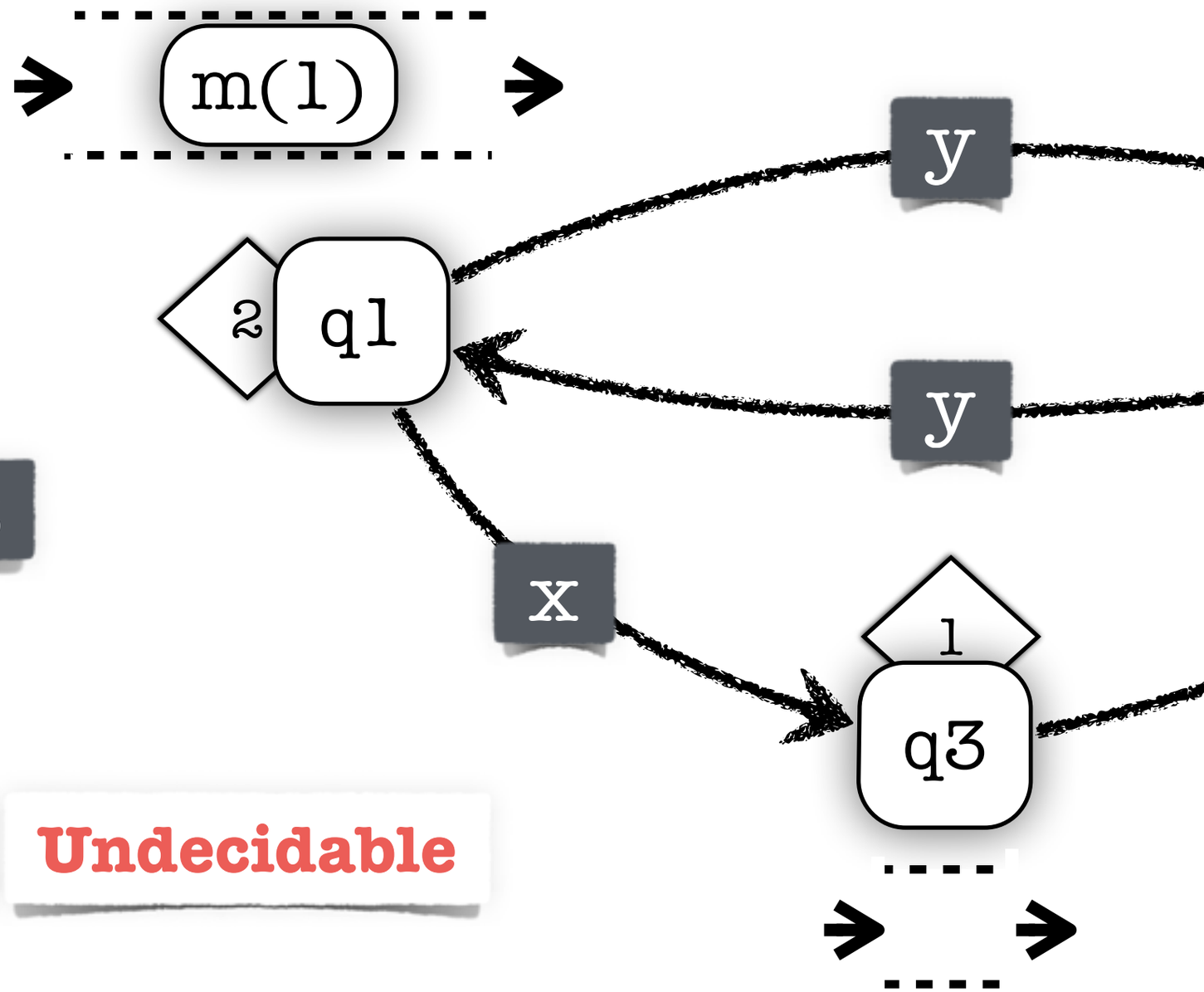
State Reachability Problem: **Undecidable**

Two counters Minsky Machine



Buffered Dynamic Register Automata

Verification of Buffered Dynamic Register Automata



Formal Model

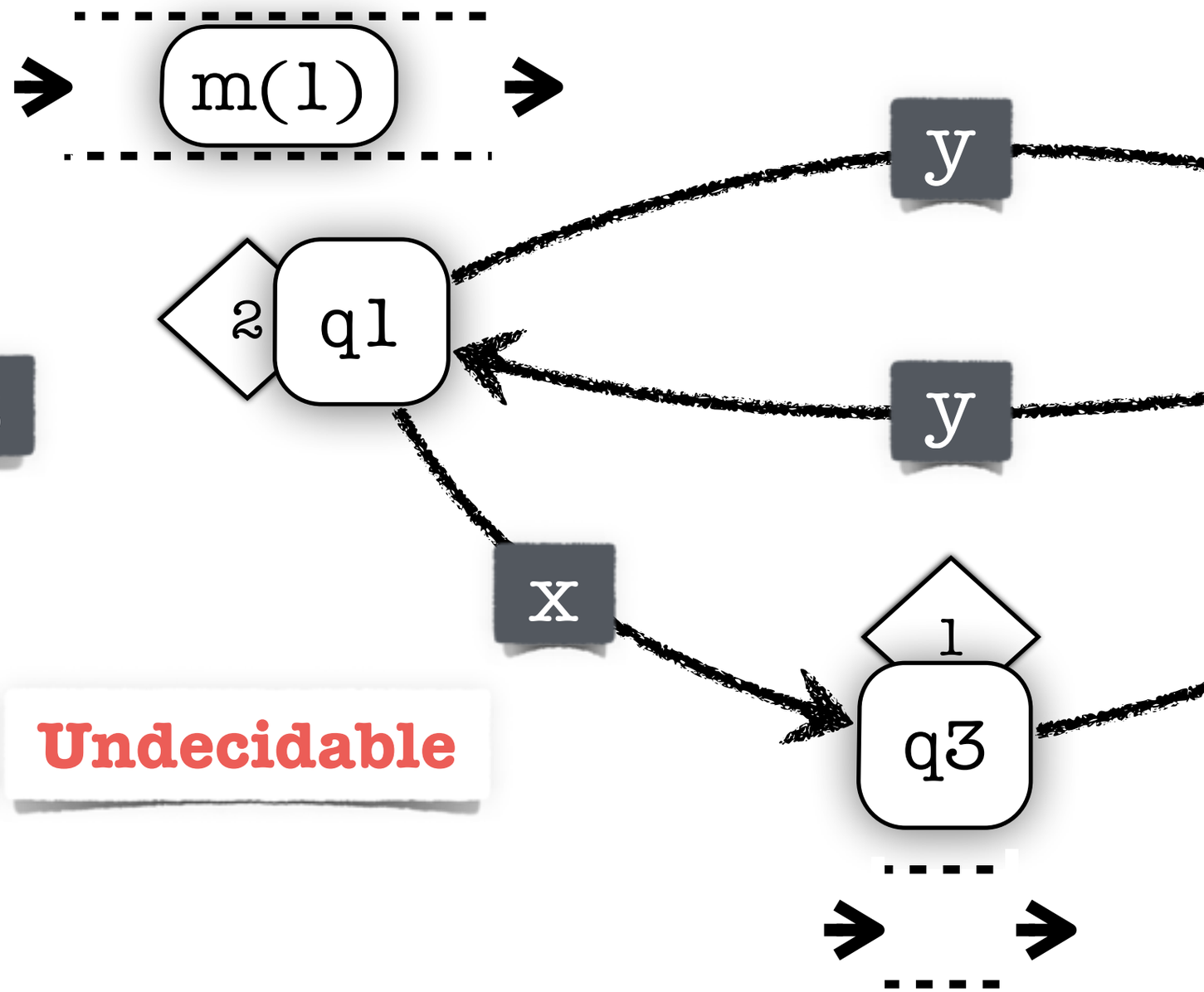
Operational Semantics

Problem:

State Reachability **Undecidable**

Buffered Dynamic Register Automata

Verification of Buffered Dynamic Register Automata



Formal Model

Operational Semantics

Problem:

State Reachability

Undecidable

Approach

Approach

Verification of Buffered Dynamic Register Automata

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Cycle

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

~~Cycle~~

...

...

...

...

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

Strongly Bound Simple Path

Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

Strongly Bound Simple Path

Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

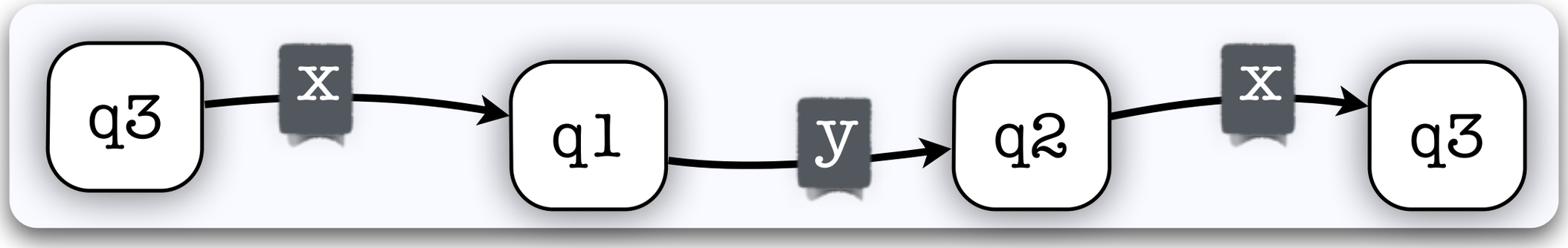
Strongly Bound Simple Path

Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Bound Simple Path

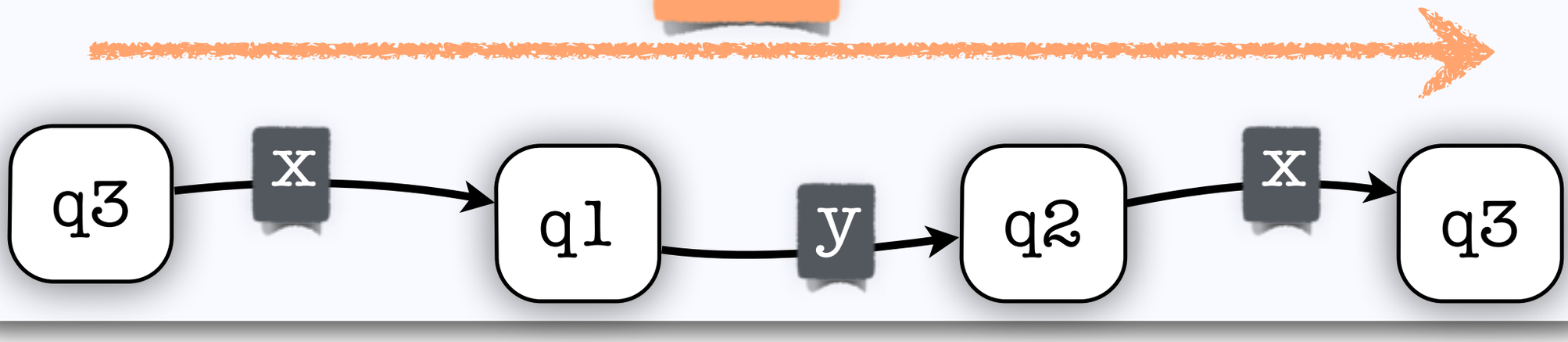


Approach

Verification of Buffered Dynamic Register Automata

Bound Simple Path

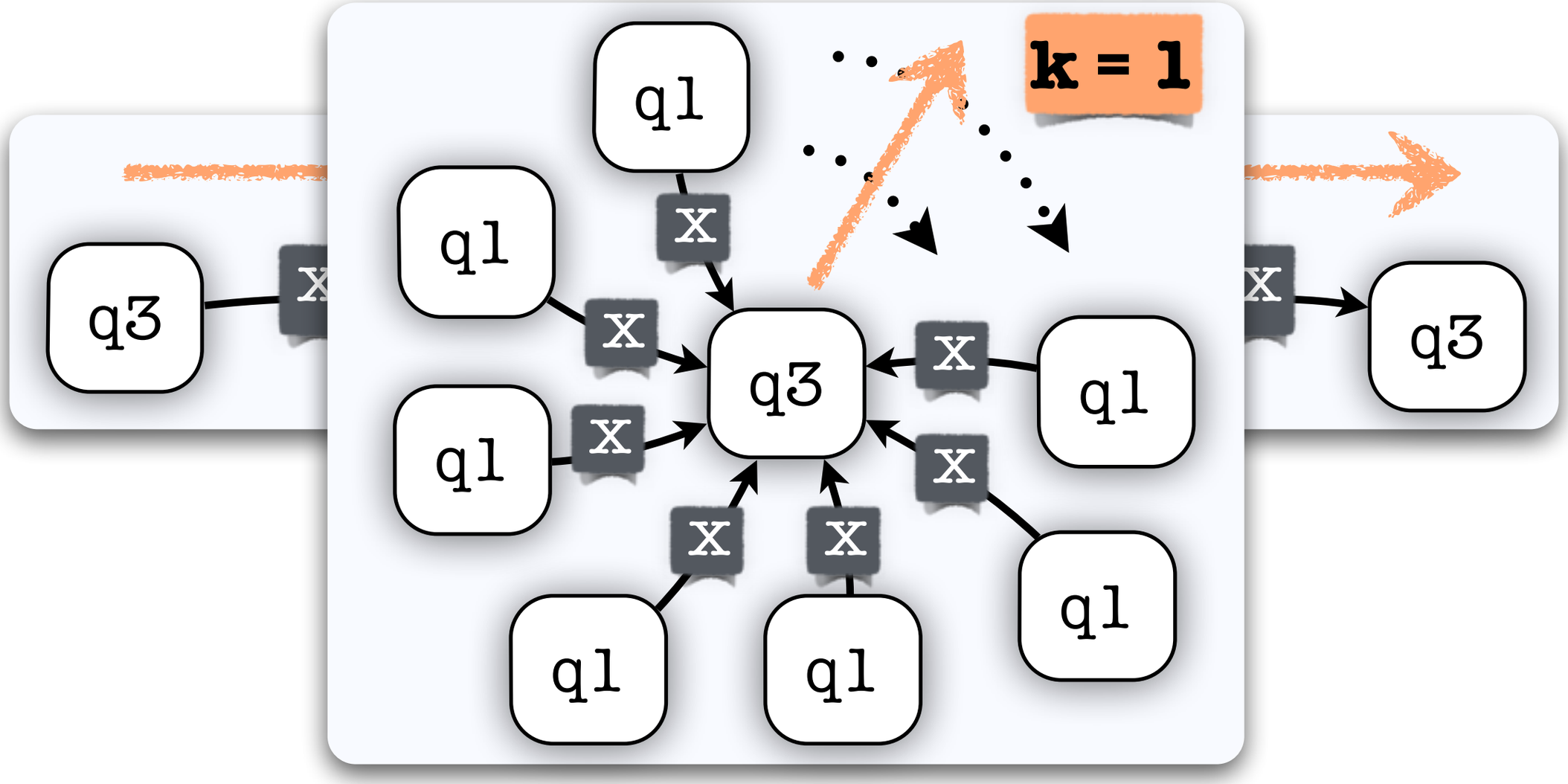
k = 3



Approach

Verification of Buffered Dynamic Register Automata

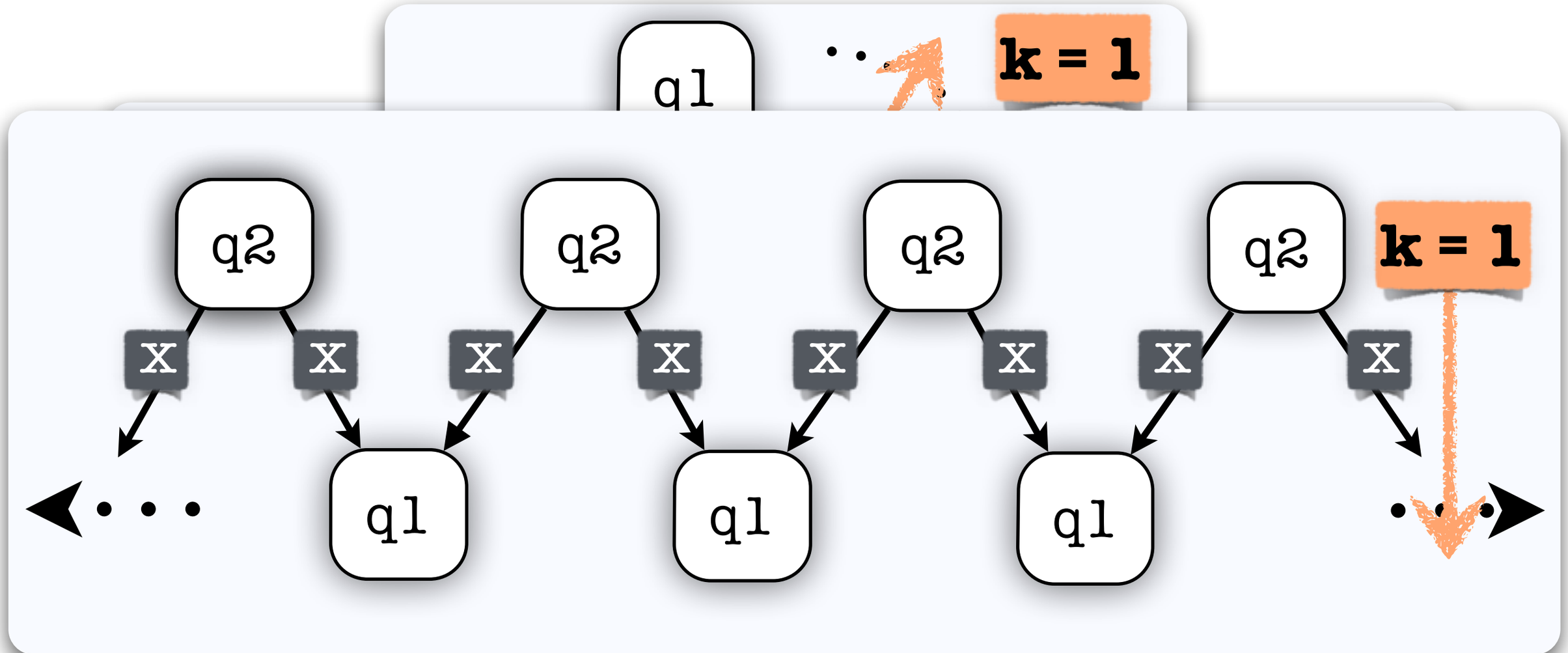
Bound Simple Path



Approach

Verification of Buffered Dynamic Register Automata

Bound Simple Path



Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

Strongly Bound Simple Path

Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

Strongly Bound Simple Path

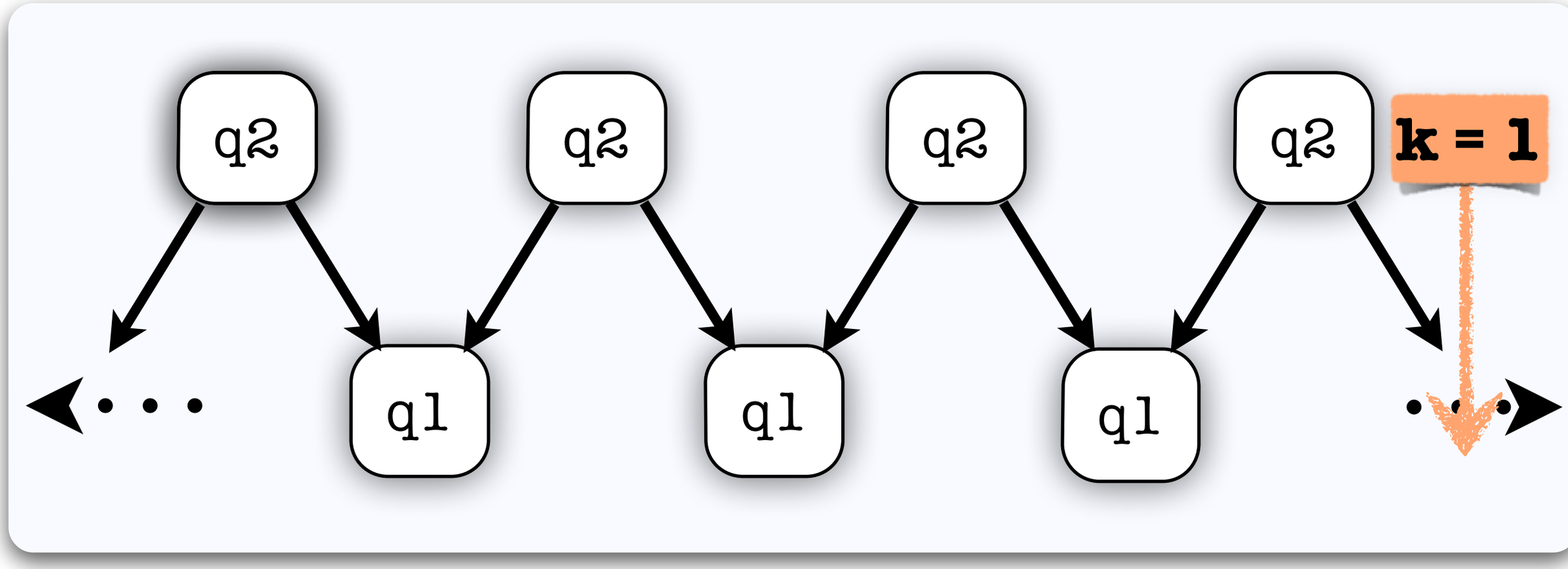
Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

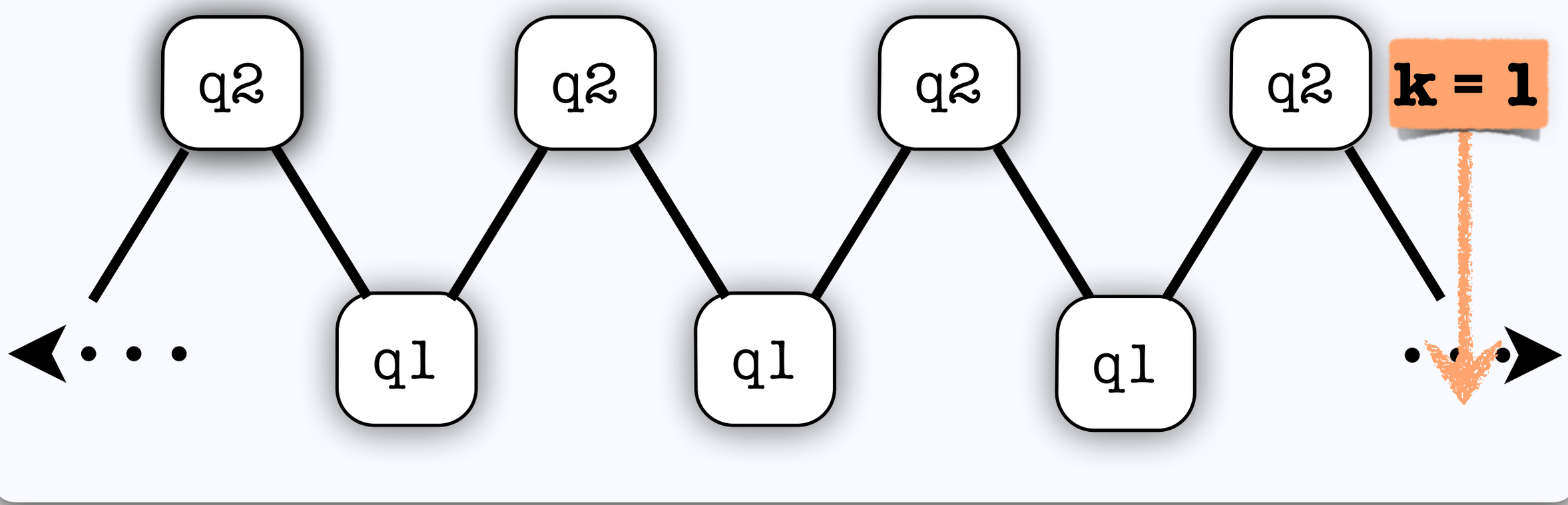
Verification of Buffered Dynamic Register Automata

Strongly Bound Simple Path



Approach

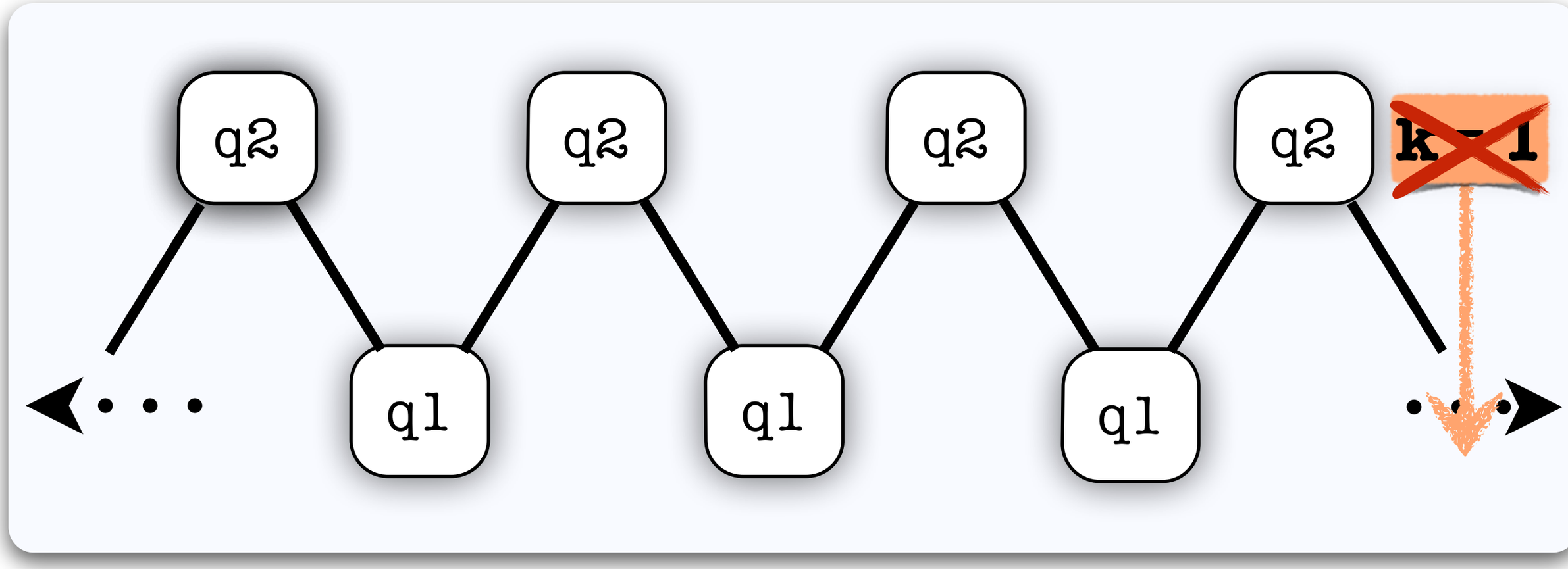
Strongly Bound Simple Path



Approach

Verification of Buffered Dynamic Register Automata

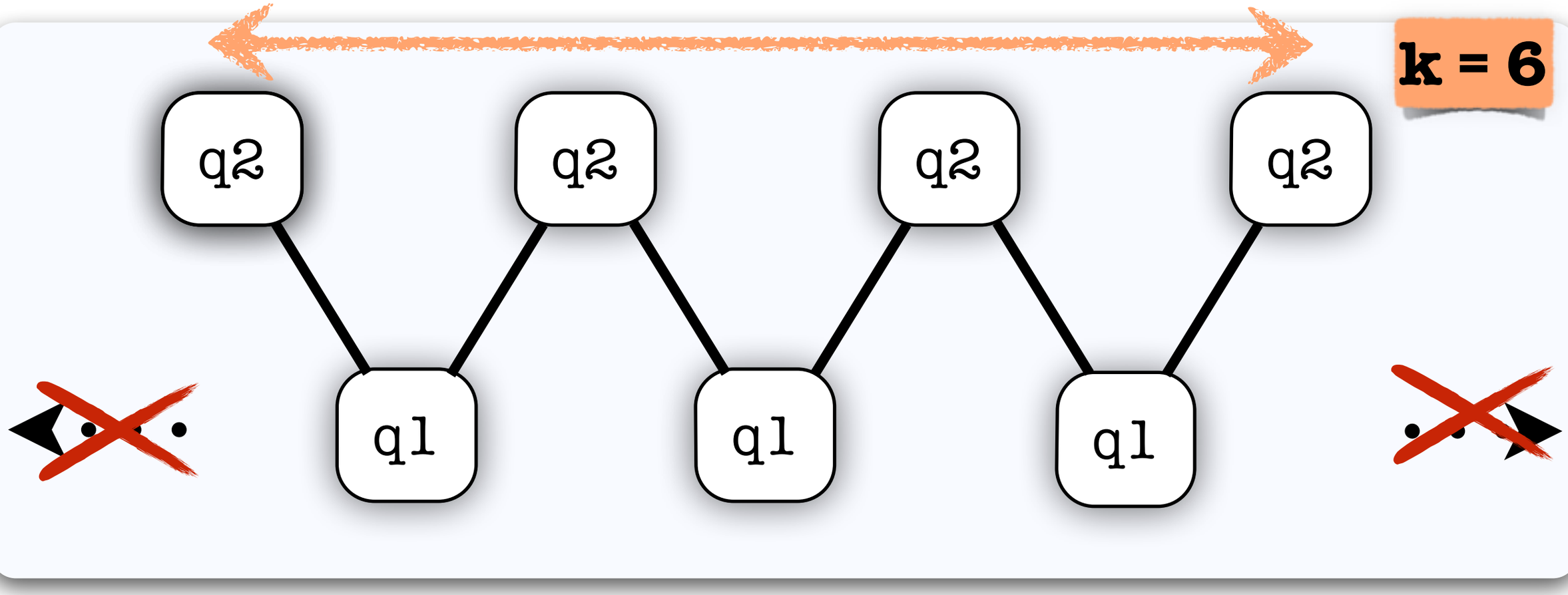
Strongly Bound Simple Path



Approach

Verification of Buffered Dynamic Register Automata

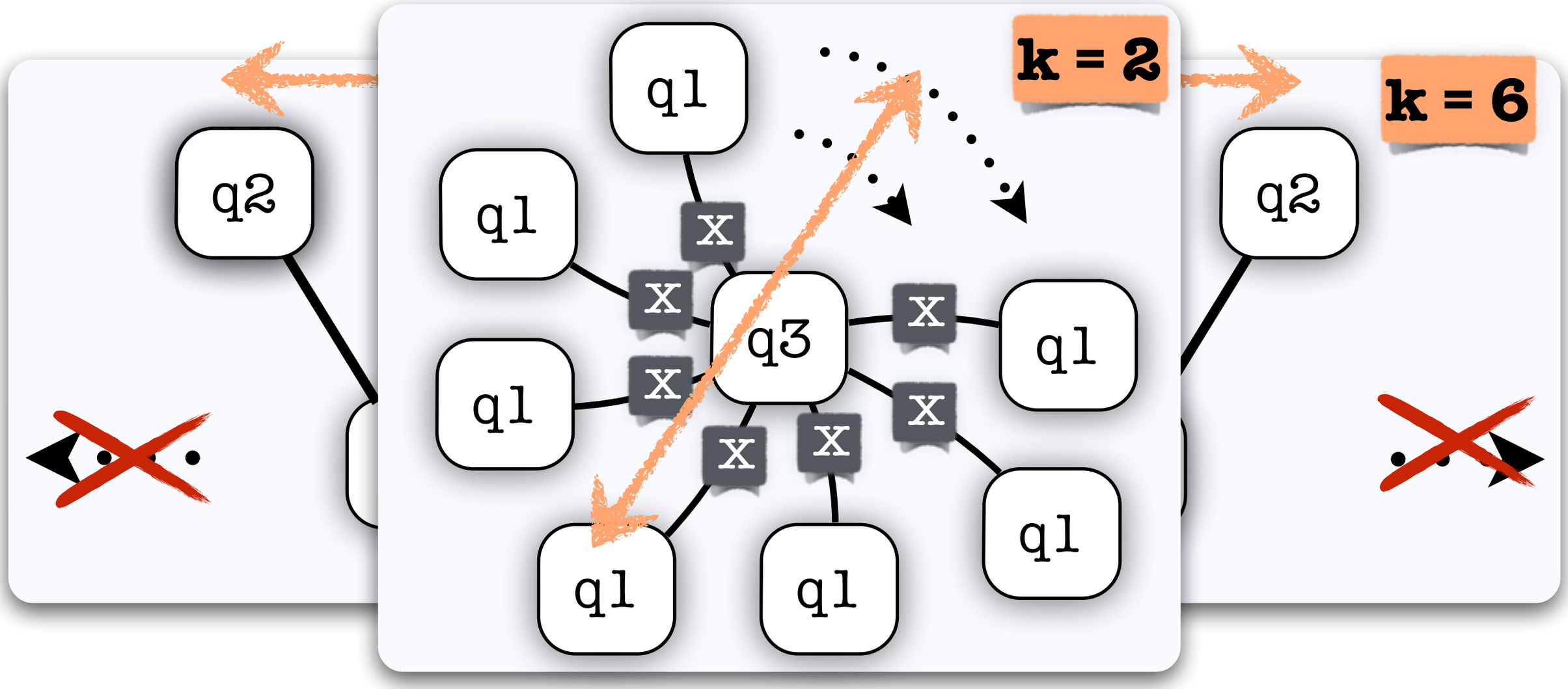
Strongly Bound Simple Path



Approach

Verification of Buffered Dynamic Register Automata

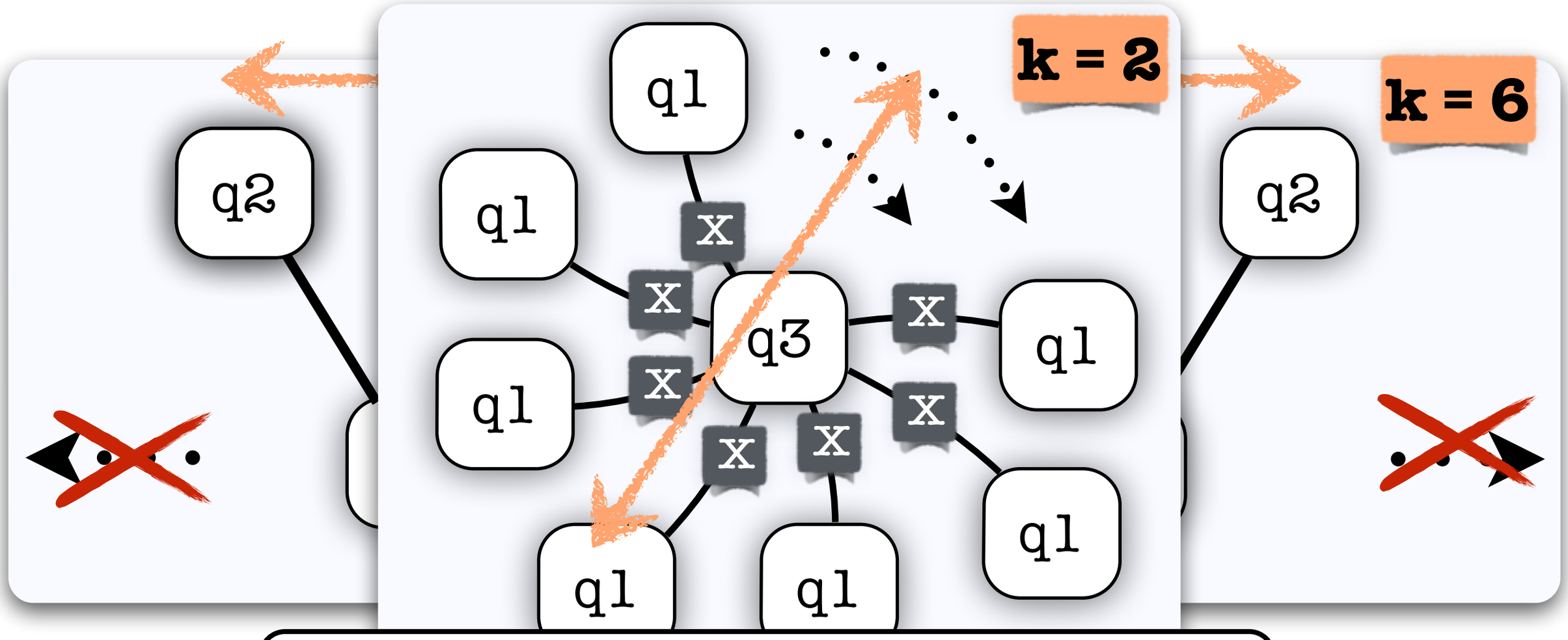
Strongly Bound Simple Path



Approach

Verification of Buffered Dynamic Register Automata

Strongly Bound Simple Path



Still: Unbounded Number of Processes

Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

Strongly Bound Simple Path

Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

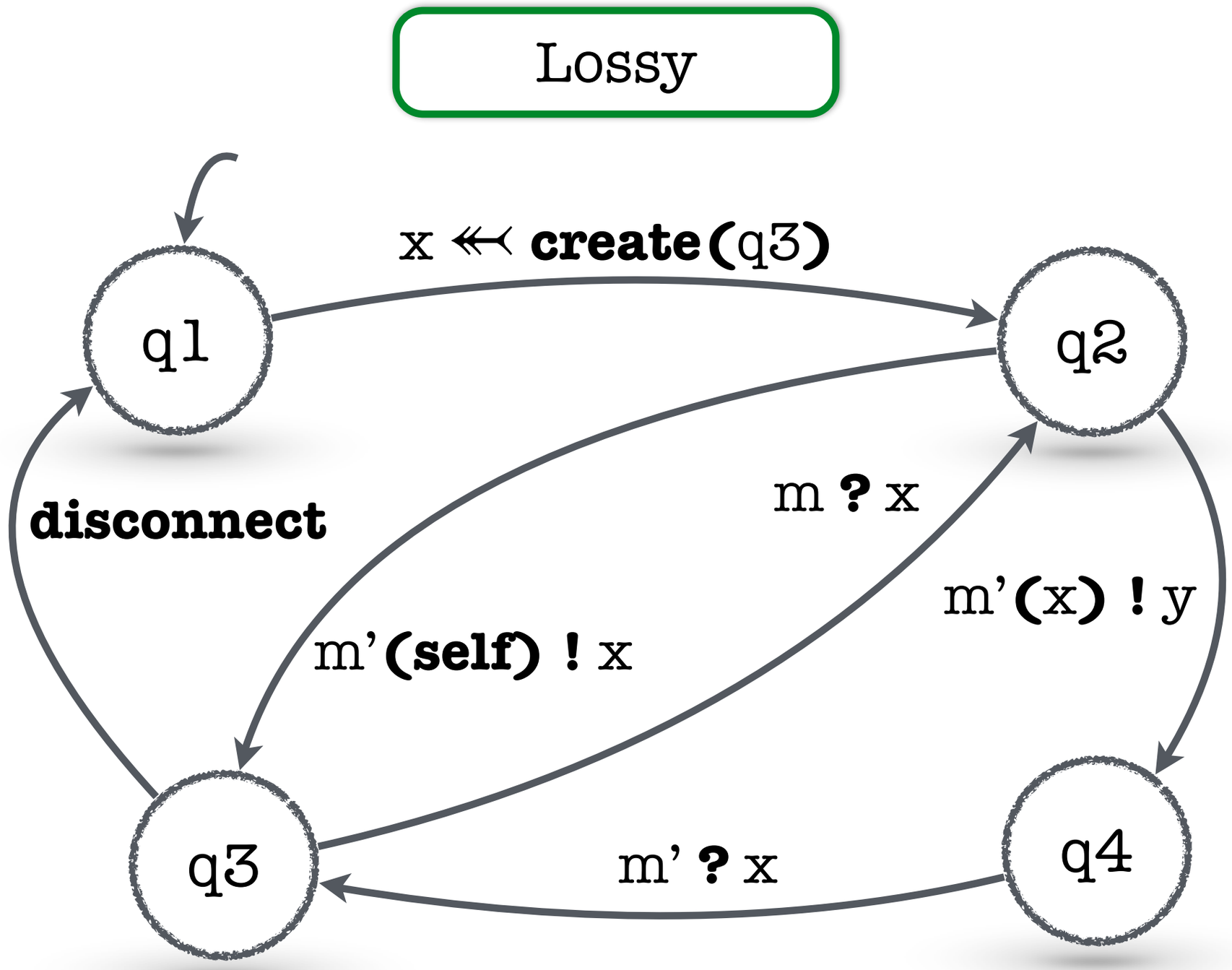
Strongly Bound Simple Path

Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

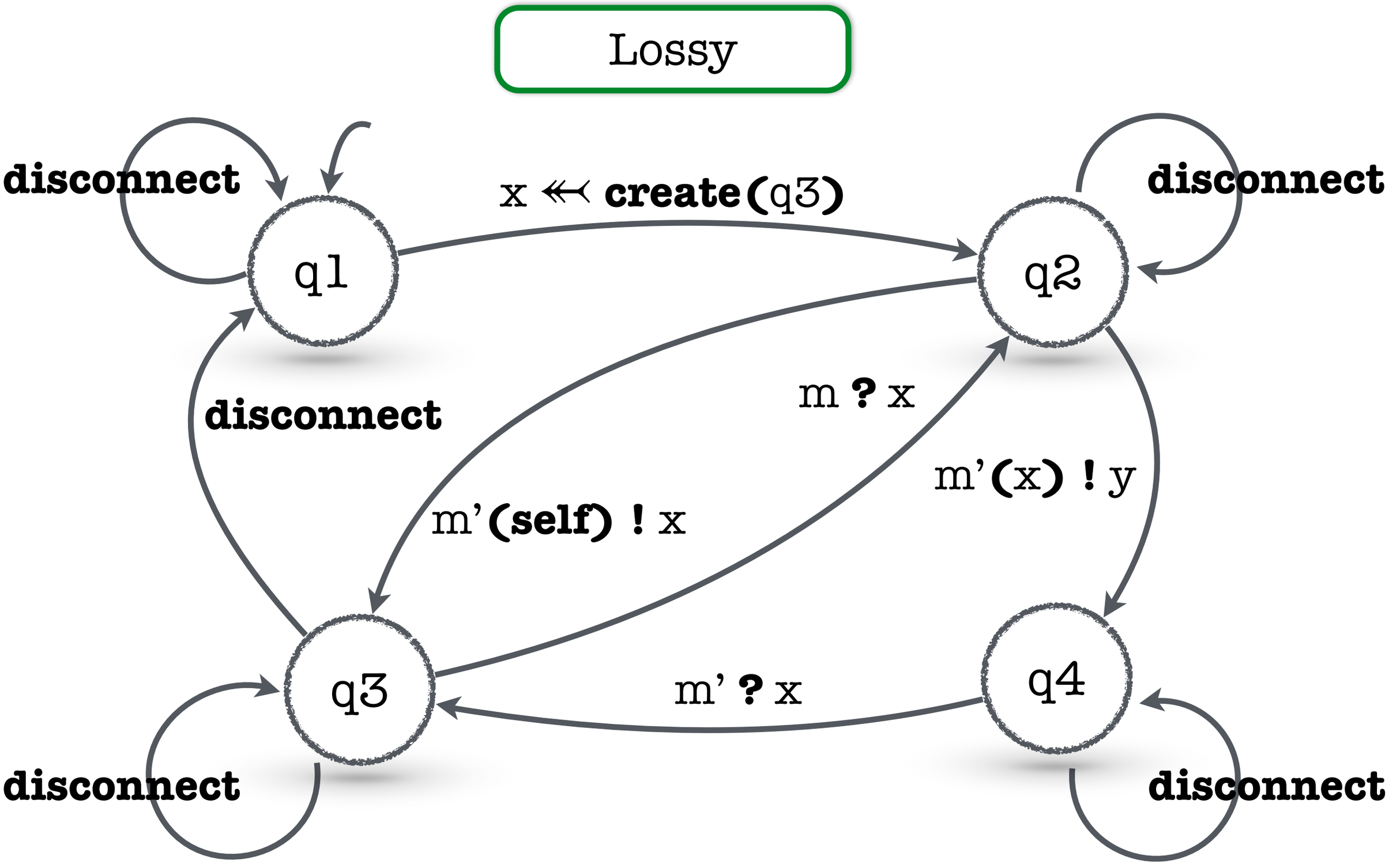
Approach

Verification of Buffered Dynamic Register Automata



Approach

Verification of Buffered Dynamic Register Automata



Approach

Verification of Buffered Dynamic Register Automata

~~Cycle~~

Bound the Buffer

Bound Simple Path

Strongly Bound Simple Path

Lossy

Define sub-classes of the system for which **state reachability** is **decidable.**

Buffered Dynamic Register Automata

Verification of Buffered Dynamic Register Automata

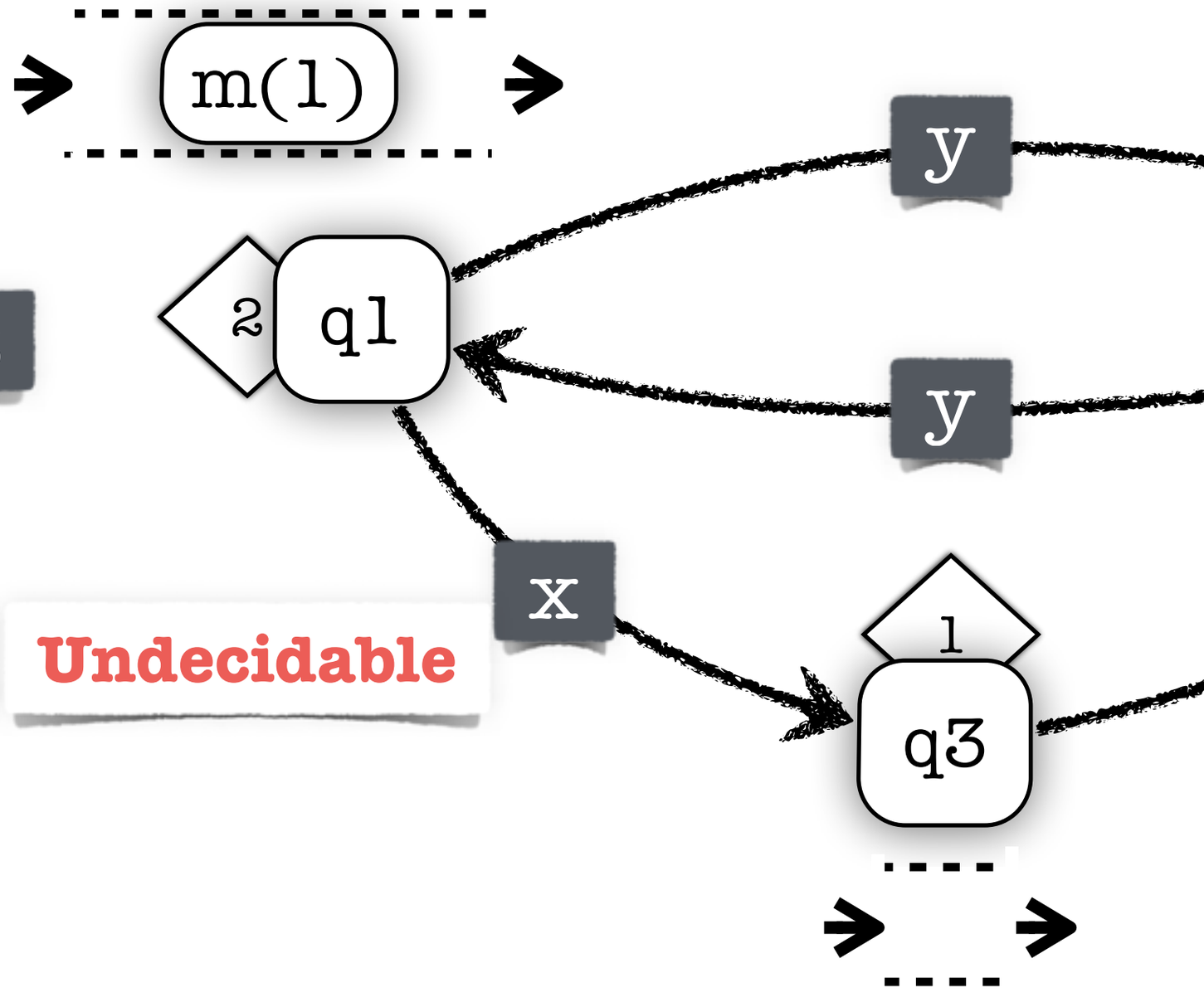
Formal Model

Operational Semantics

Problem:

State Reachability — **Undecidable**

Approach



Buffered Dynamic Register Automata

Formal Model

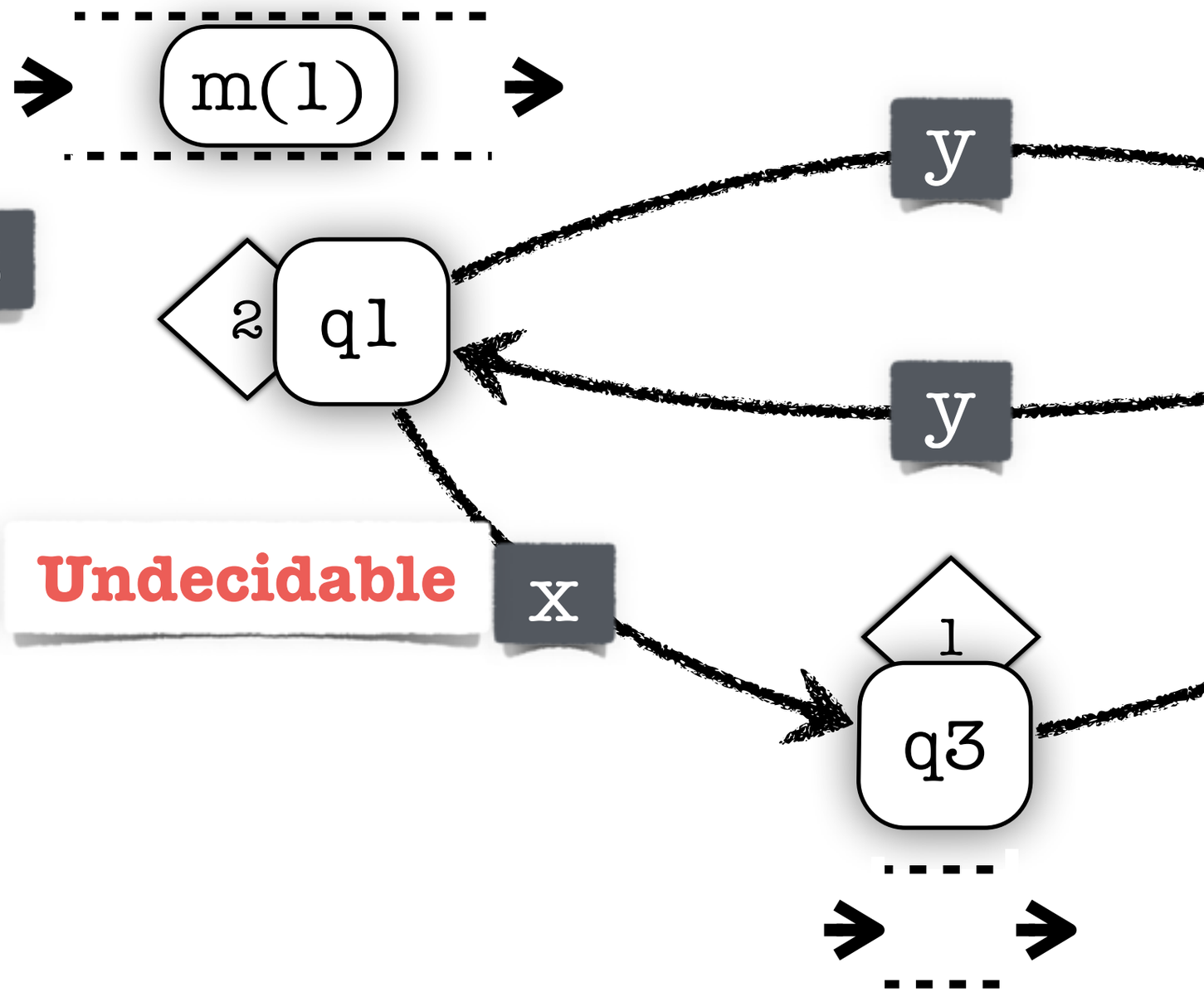
Operational Semantics

Problem:

State Reachability **Undecidable**

Approach

Results



Results

State Reachability

Results

Verification of Buffered Dynamic Register Automata

State Reachability

~~. Cycles .~~

Results

Verification of Buffered Dynamic Register Automata

State Reachability

~~Cycles~~

Results

Verification of Buffered Dynamic Register Automata

State Reachability

~~Cycles~~

Strongly Bound Simple Path

Results

Verification of Buffered Dynamic Register Automata

State Reachability

~~. Cycles .~~

Strongly Bound Simple Path

Bound Buffers

Results

State Reachability

~~. Cycles .~~	+	Strongly Bound Simple Path
~~. Cycles .~~	+	Bound Buffers
Bound Buffers	+	Strongly Bound Simple Path

Results

Verification of Buffered Dynamic Register Automata

State Reachability

~~. Cycles .~~ + Strongly Bound Simple Path

~~. Cycles .~~ + Bound Buffers

Bound Buffers + Strongly Bound Simple Path

~~.Cycles.~~ + Bound Buffers + Strongly Bound Simple Path

Results

Verification of Buffered Dynamic Register Automata

State Reachability

~~. Cycles .~~ + Strongly Bound Simple Path

~~. Cycles .~~ + Bound Buffers

Bound Buffers + Strongly Bound Simple Path

~~.Cycles.~~ + Bound Buffers + Strongly Bound Simple Path

Bound Buffers Strongly Bound Simple Path

Results

Verification of Buffered Dynamic Register Automata

State Reachability

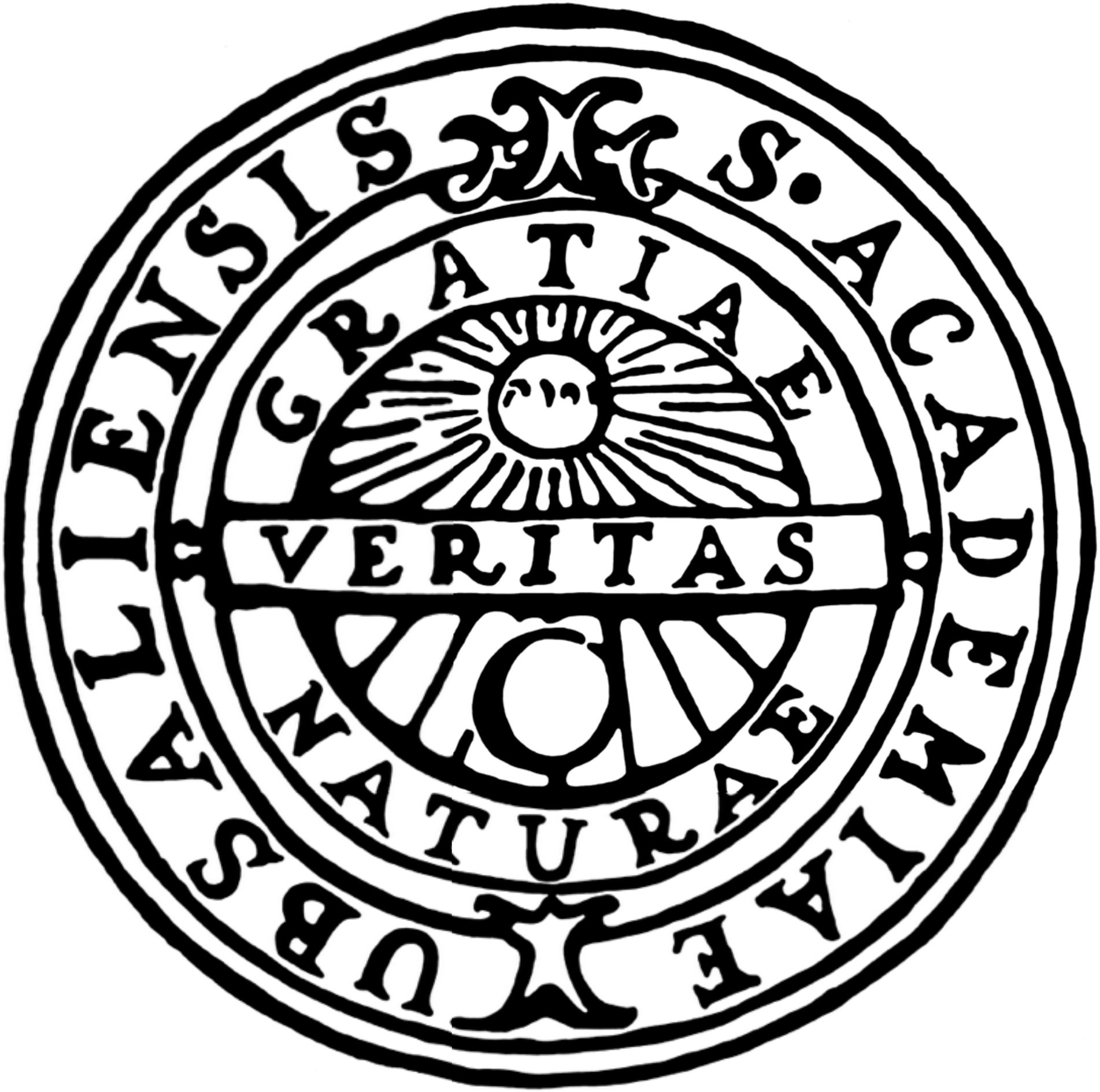
~~. Cycles .~~ + Strongly Bound Simple Path

~~. Cycles .~~ + Bound Buffers

Bound Buffers + Strongly Bound Simple Path

~~.Cycles.~~ + Bound Buffers + Strongly Bound Simple Path

Lossy + Bound Buffers + Strongly Bound Simple Path


S·ACAD
GRATIAE
VERITAS
NATURAE
OBSERVATORIUM

Proofs

Verification of Buffered Dynamic Register Automata

~~Cycles~~ + Strongly Bound Simple Path

~~Cycles~~ + Bound Buffers

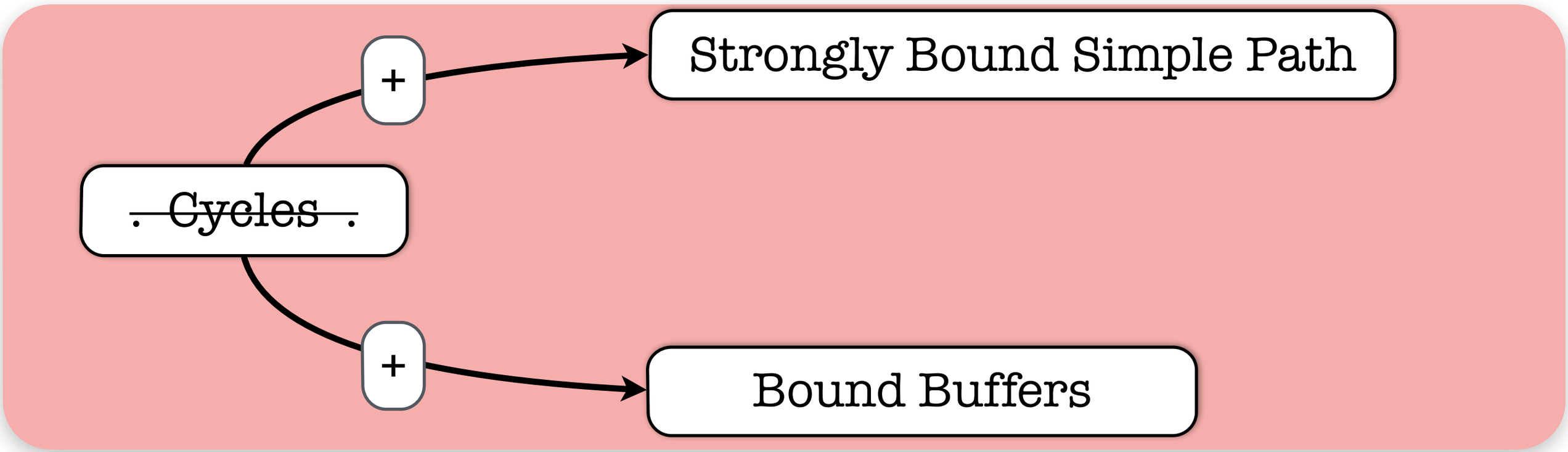
Bound Buffers + Strongly Bound Simple Path

Acyclic + Bound the Buffer + Strongly Bound Simple Path

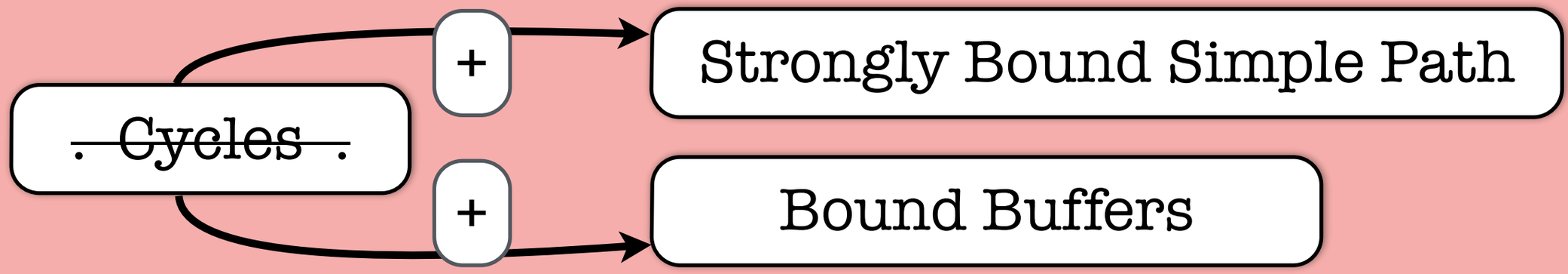
Lossy + Bound the Buffer + Strongly Bound Simple Path

Proofs

Verification of Buffered Dynamic Register Automata



Proofs



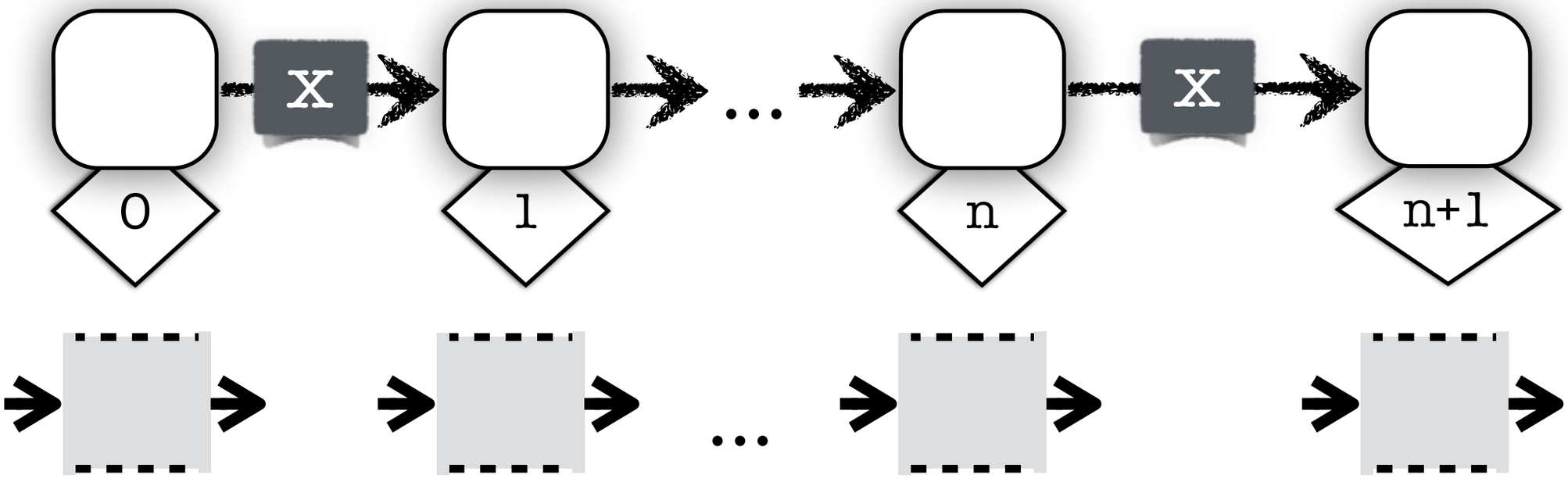
Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathrm{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in \mathrm{L}(\mathbf{B})$.

Proofs

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in \mathrm{L}(\mathbf{B})$.

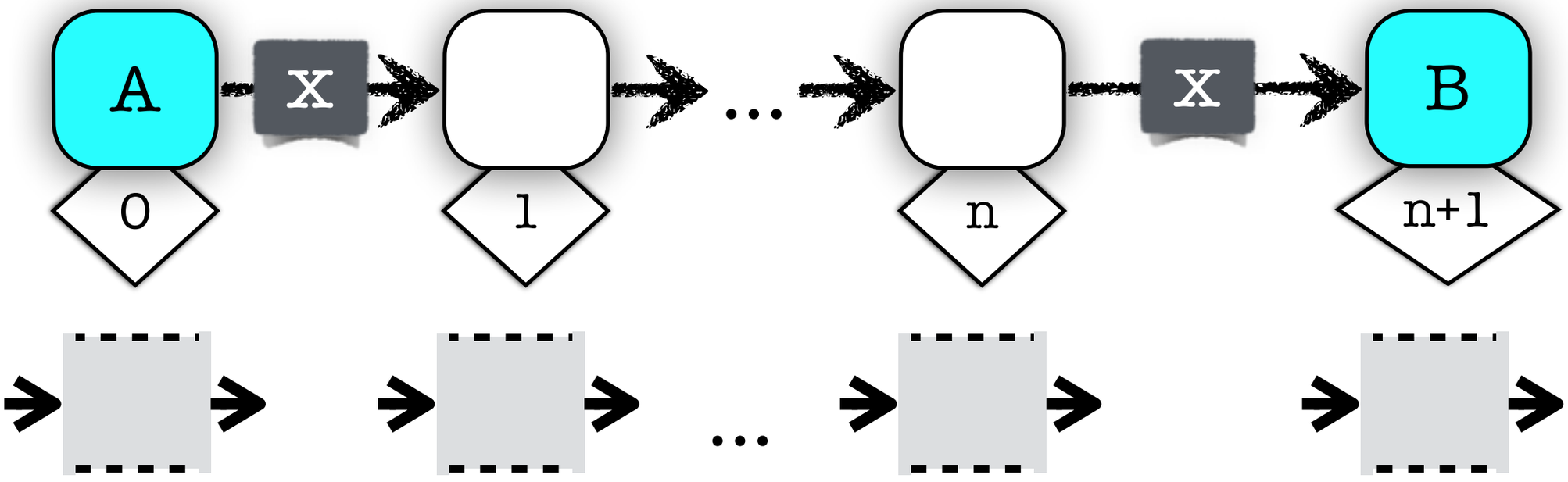


Proofs

Verification of Buffered Dynamic Register Automata

Transduction Problem

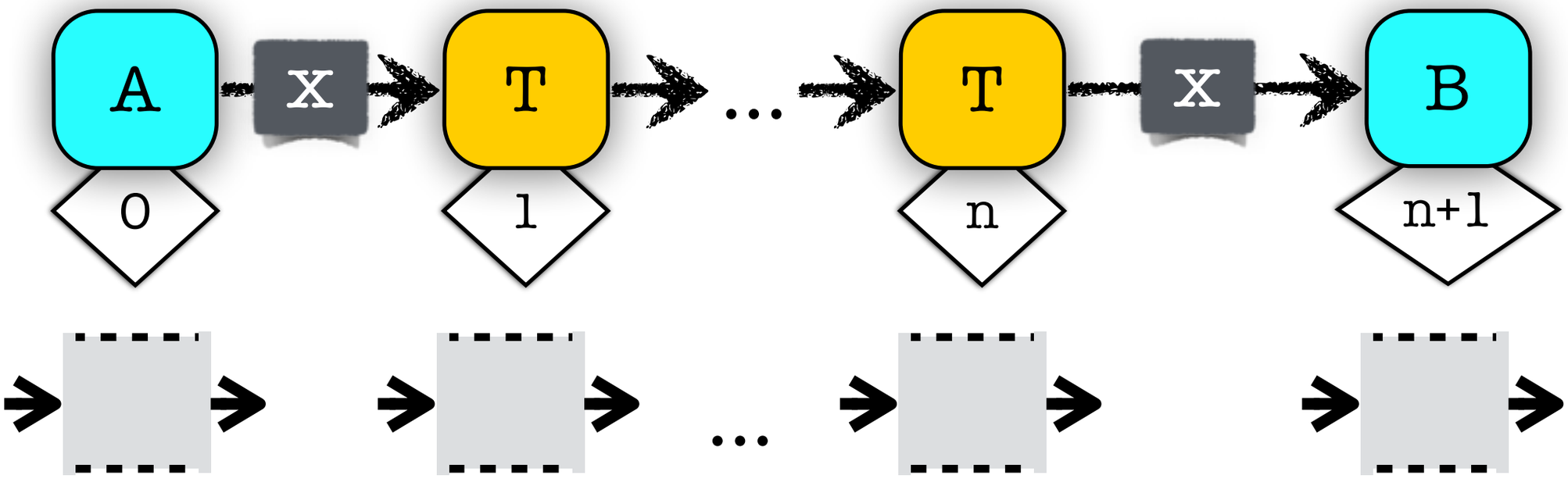
Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in \mathrm{L}(\mathbf{B})$.



Proofs

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathrm{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in \mathrm{L}(\mathbf{B})$.

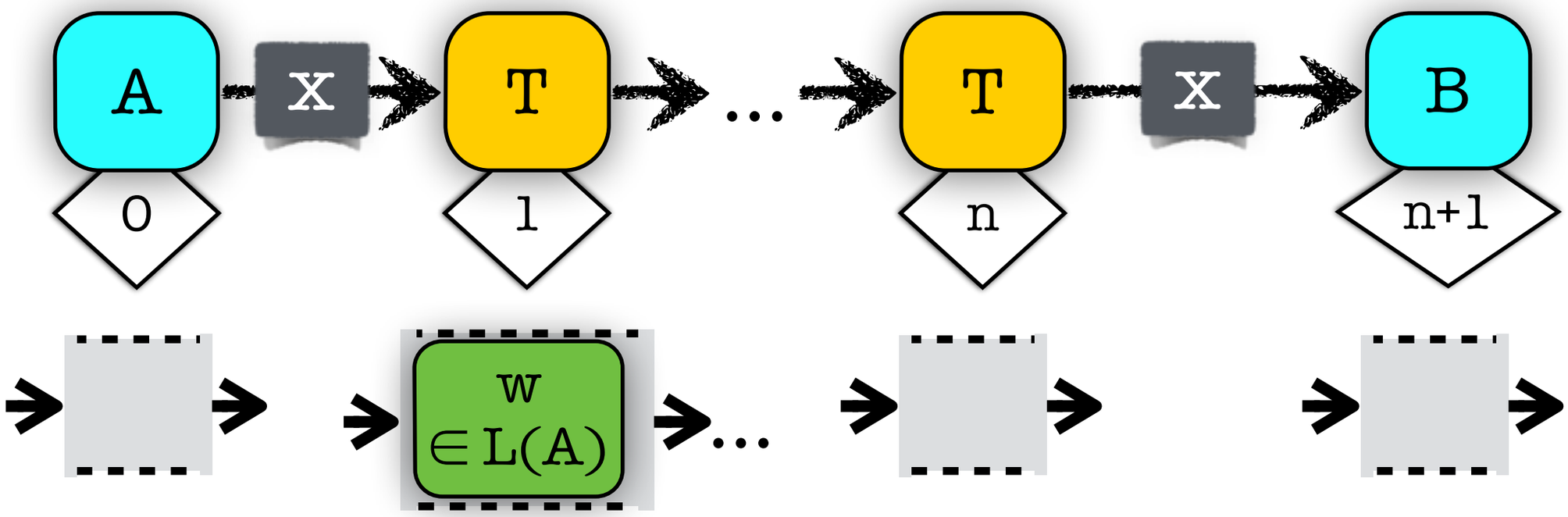


Proofs

Verification of Buffered Dynamic Register Automata

Transduction Problem

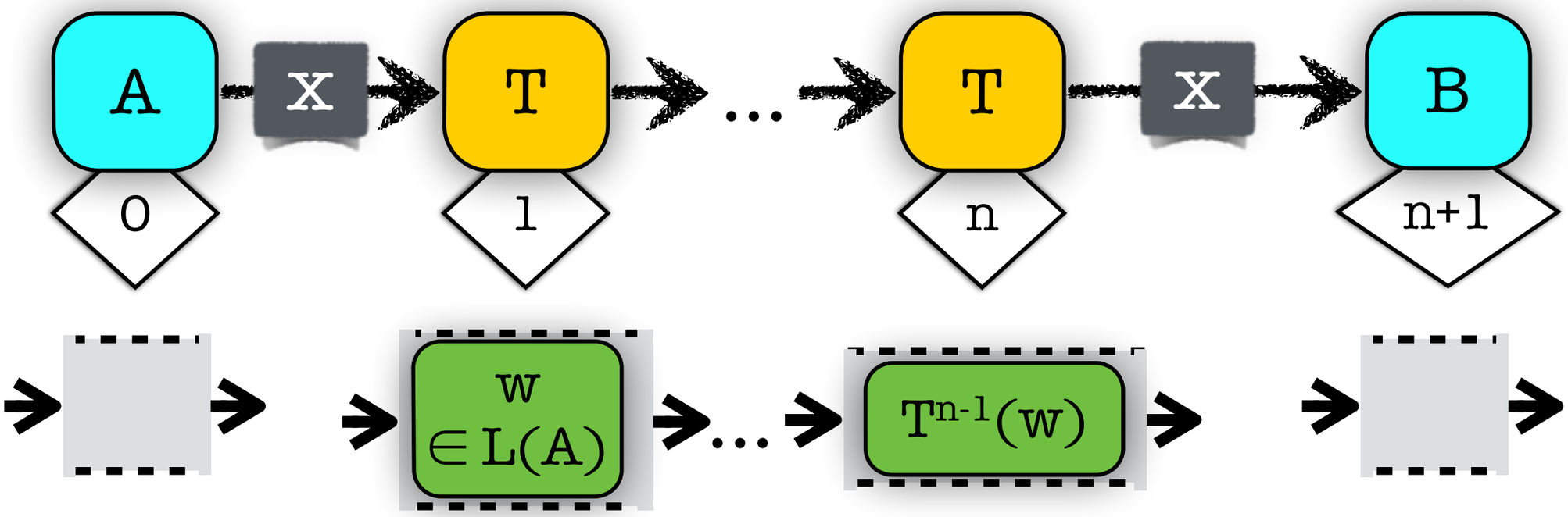
Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in L(\mathbf{B})$.



Proofs

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in L(\mathbf{B})$.

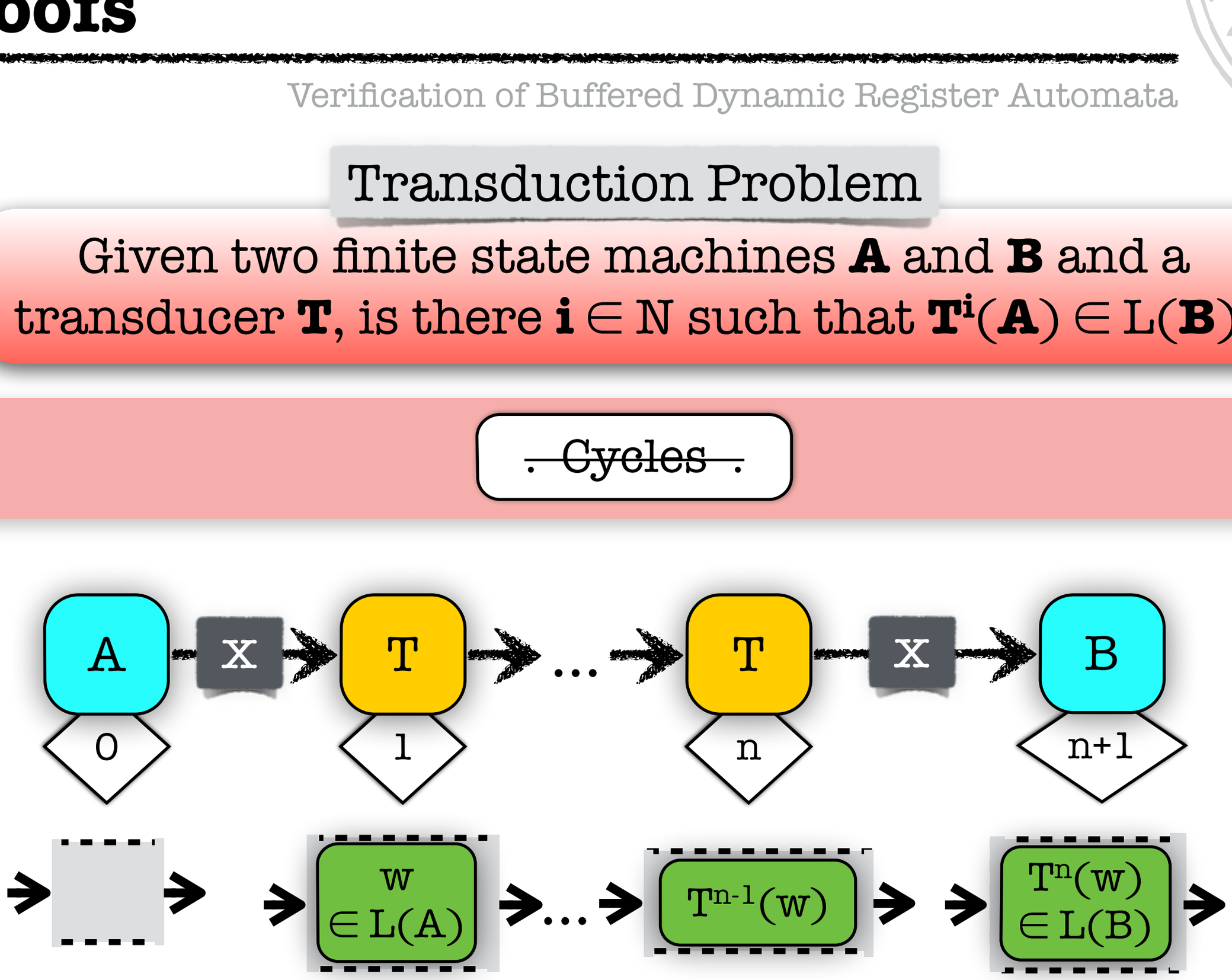


Proofs

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in L(\mathbf{B})$.

~~Cycles~~

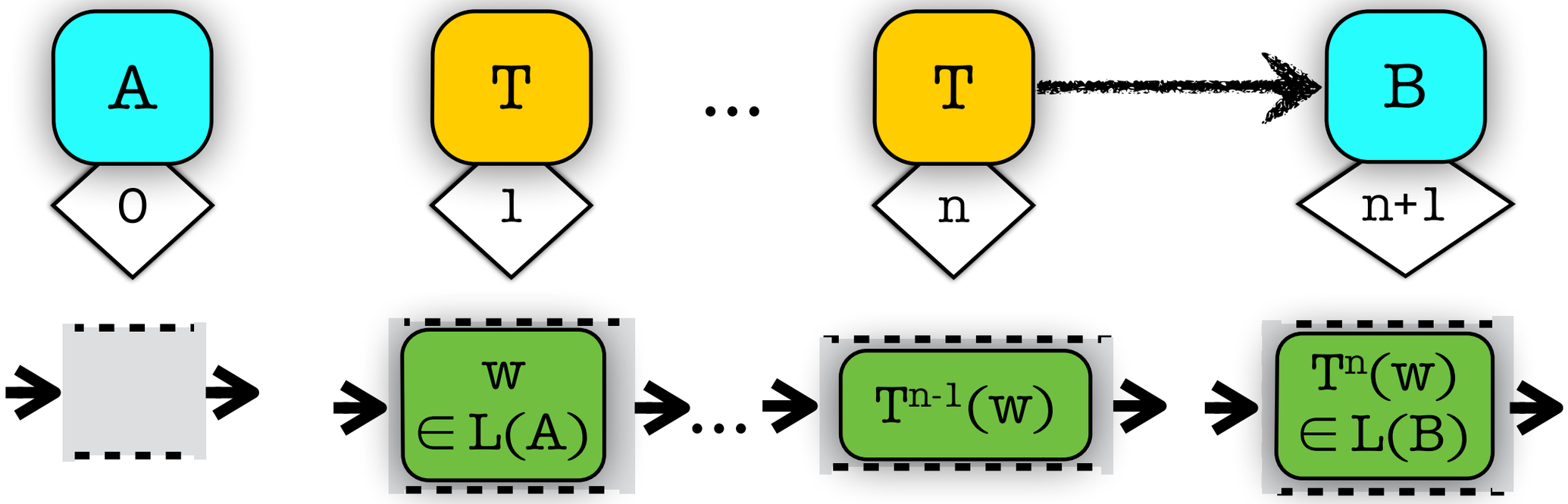


Proofs

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in L(\mathbf{B})$.

~~Cycles~~ | Bound Simple Path

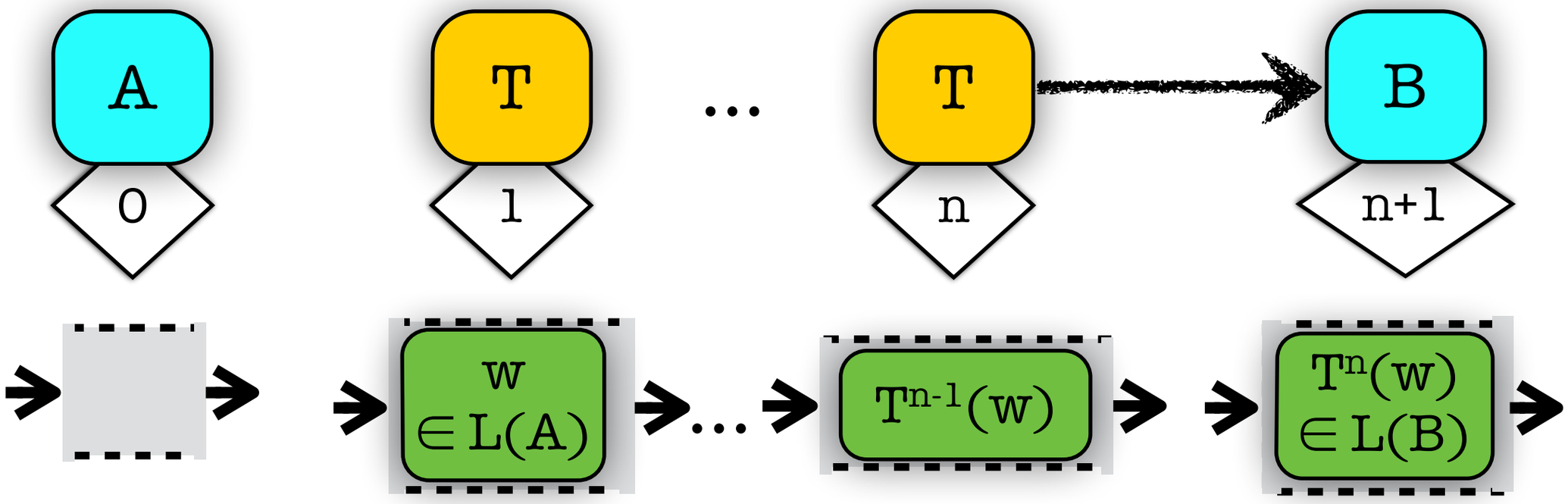


Proofs

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in L(\mathbf{B})$.

~~. Cycles .~~ | Strongly Bound Simple Path



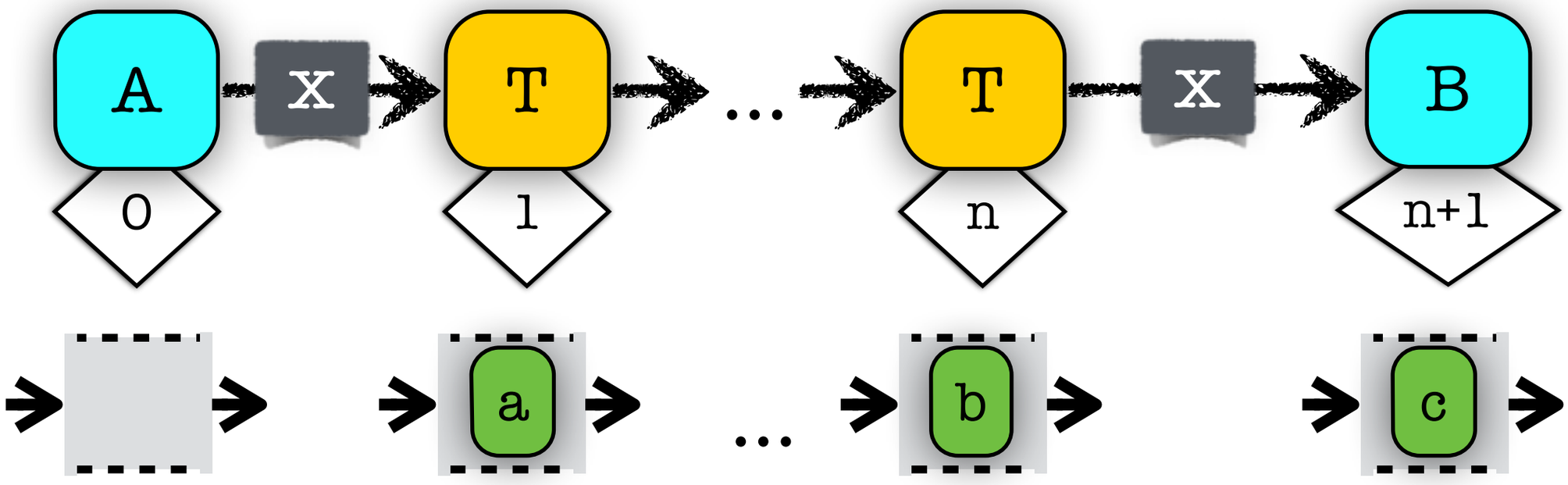
Proofs

Verification of Buffered Dynamic Register Automata

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in \mathrm{L}(\mathbf{B})$.

~~Cycles~~



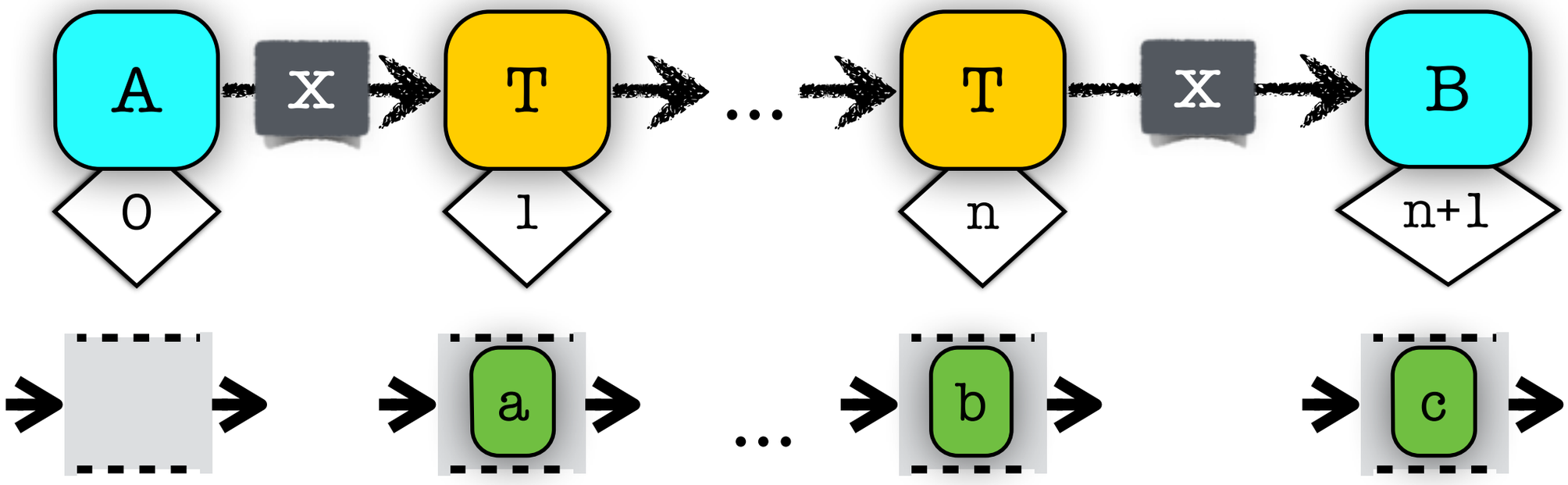
Proofs

Verification of Buffered Dynamic Register Automata

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in \mathrm{L}(\mathbf{B})$.

~~Cycles~~ | Bounded Buffers



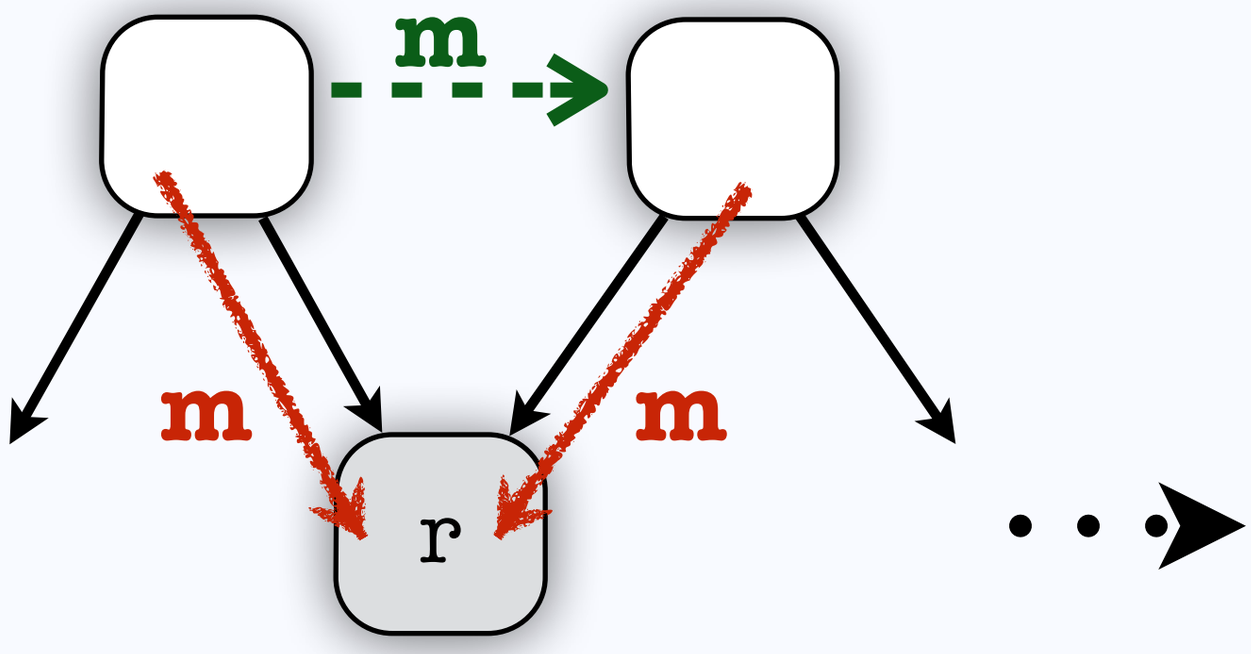
Proofs

Verification of Buffered Dynamic Register Automata

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in L(\mathbf{B})$.

~~Cycles~~ | Bounded Buffers | Bound Simple Path

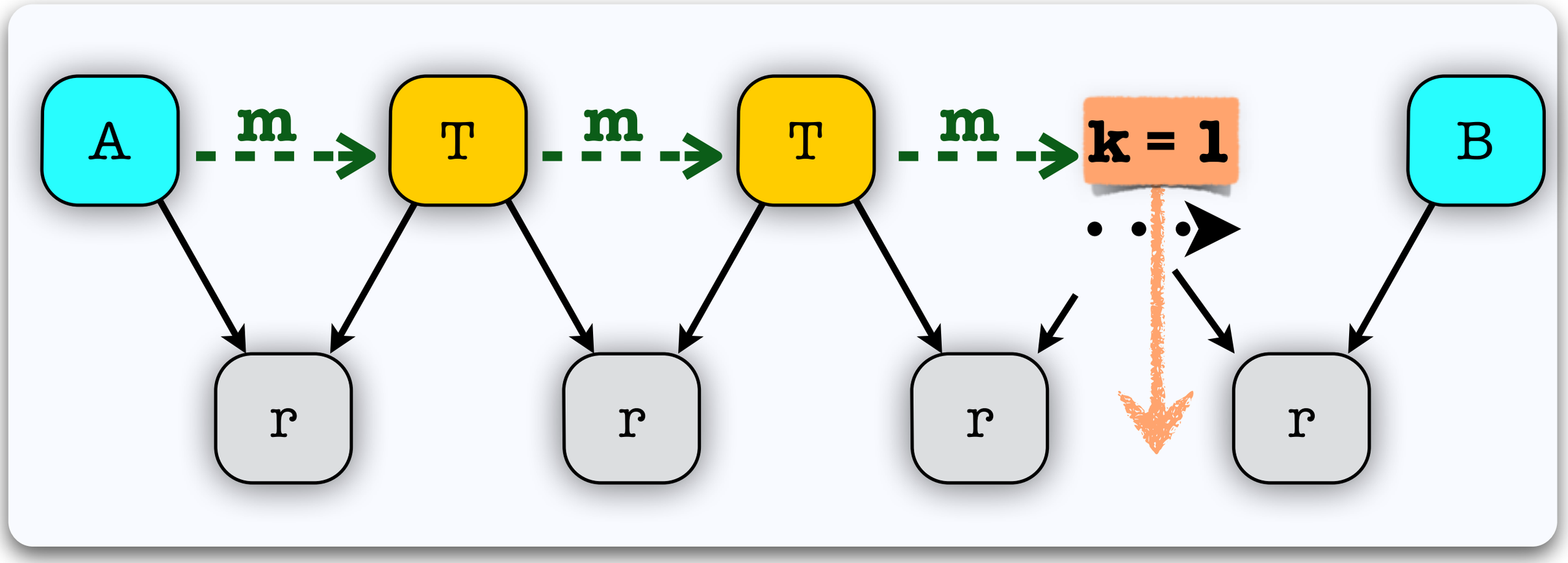


Proofs

Transduction Problem

Given two finite state machines **A** and **B** and a transducer **T**, is there $\mathbf{i} \in \mathbb{N}$ such that $\mathbf{T}^{\mathbf{i}}(\mathbf{A}) \in \mathrm{L}(\mathbf{B})$.

~~Cycles~~ | Bounded Buffers | Bound Simple Path



Proofs

Verification of Buffered Dynamic Register Automata

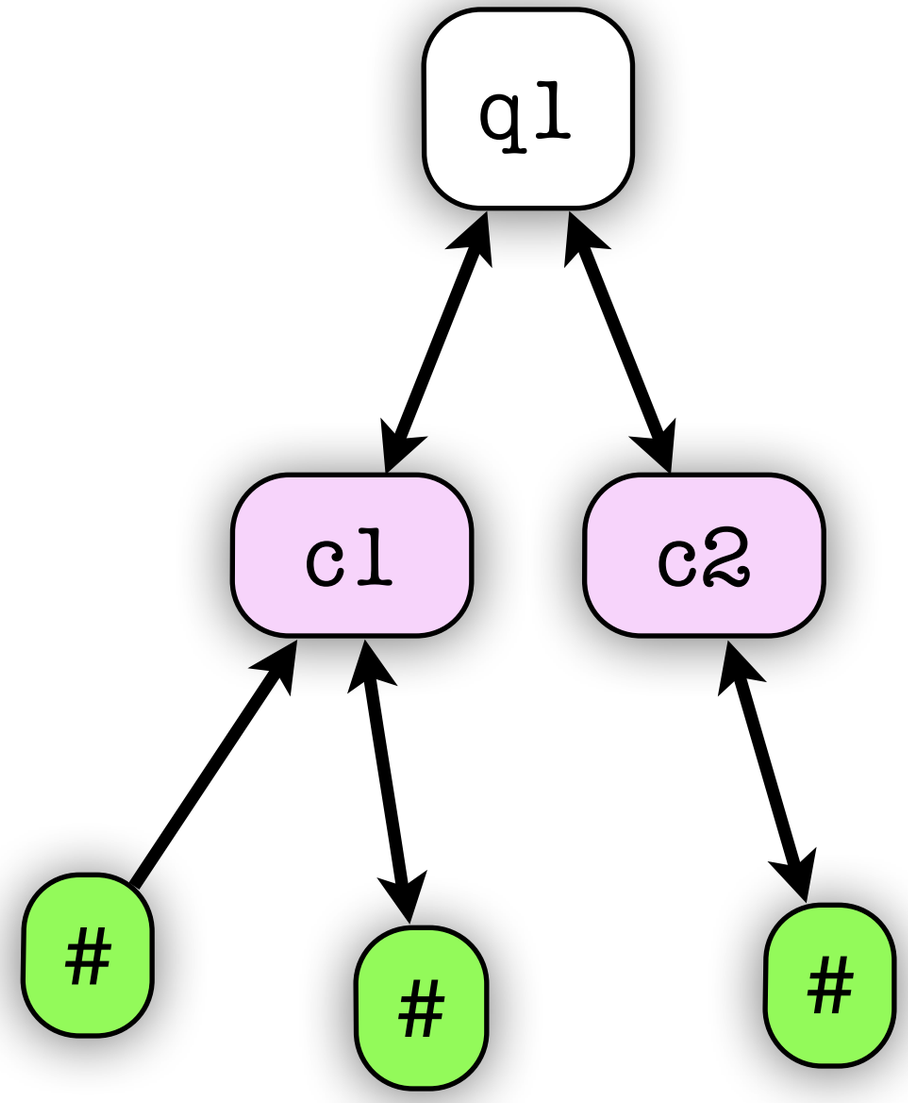
Bound Buffers + Strongly Bounded Simple Path

Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

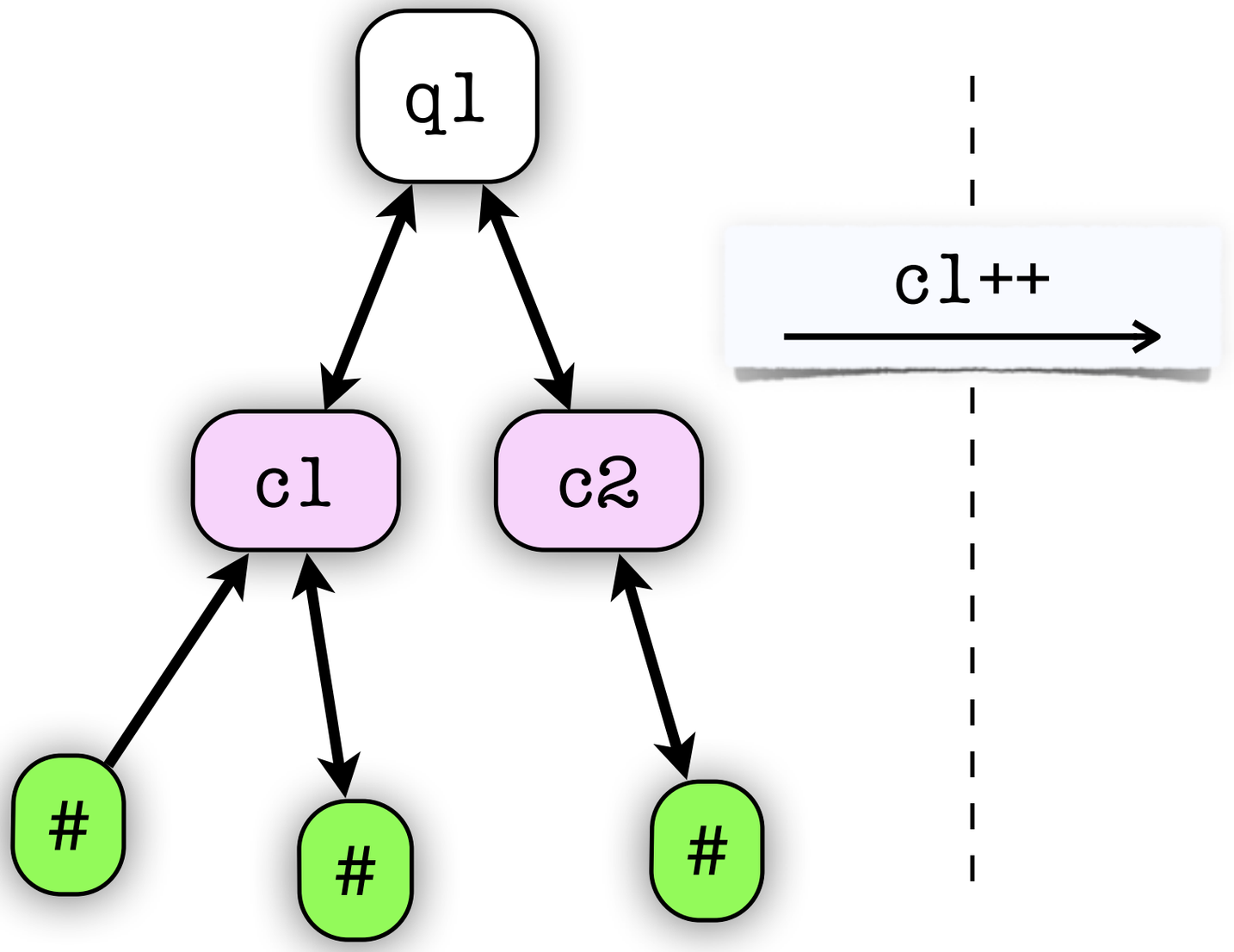


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

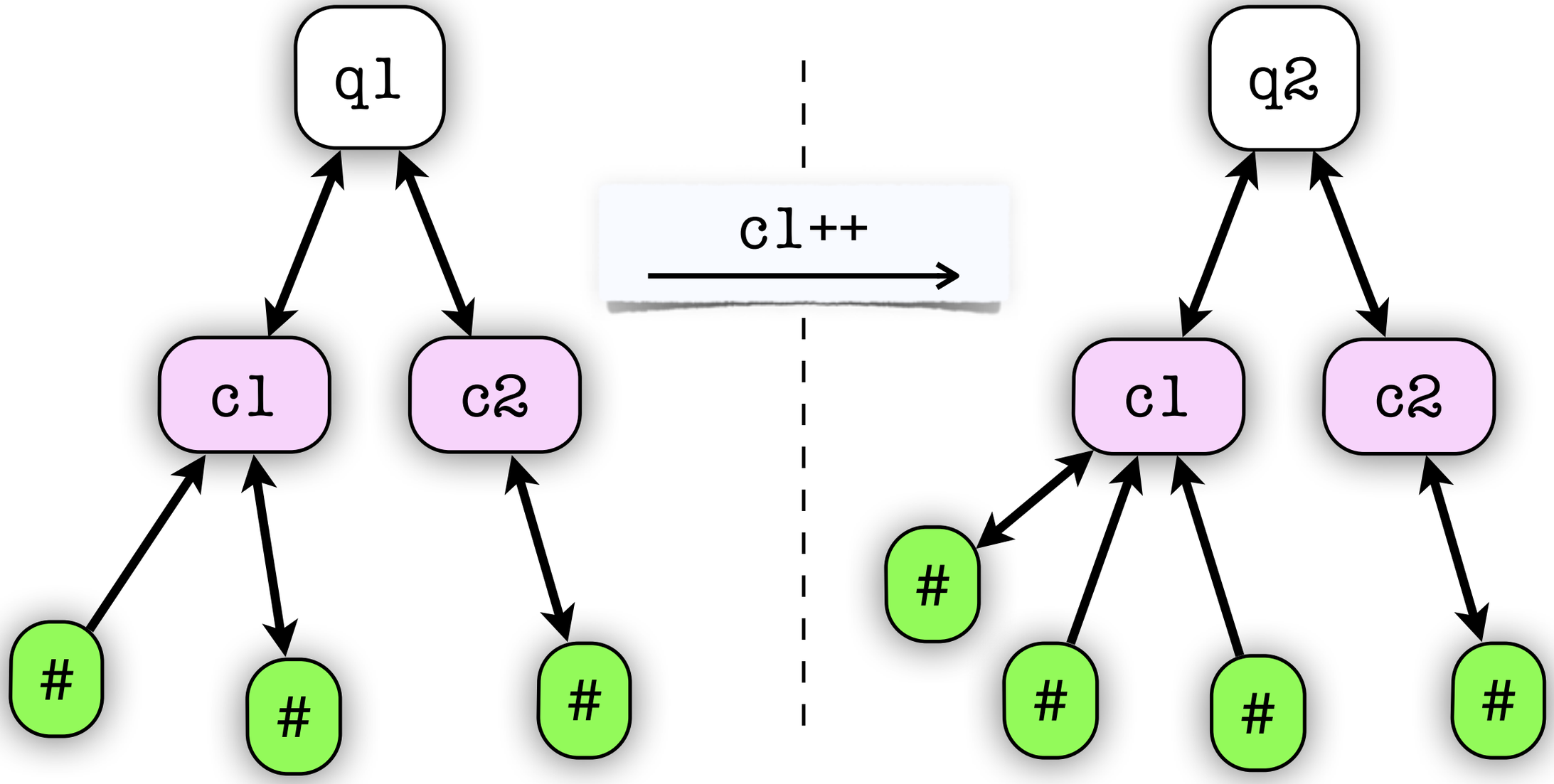


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

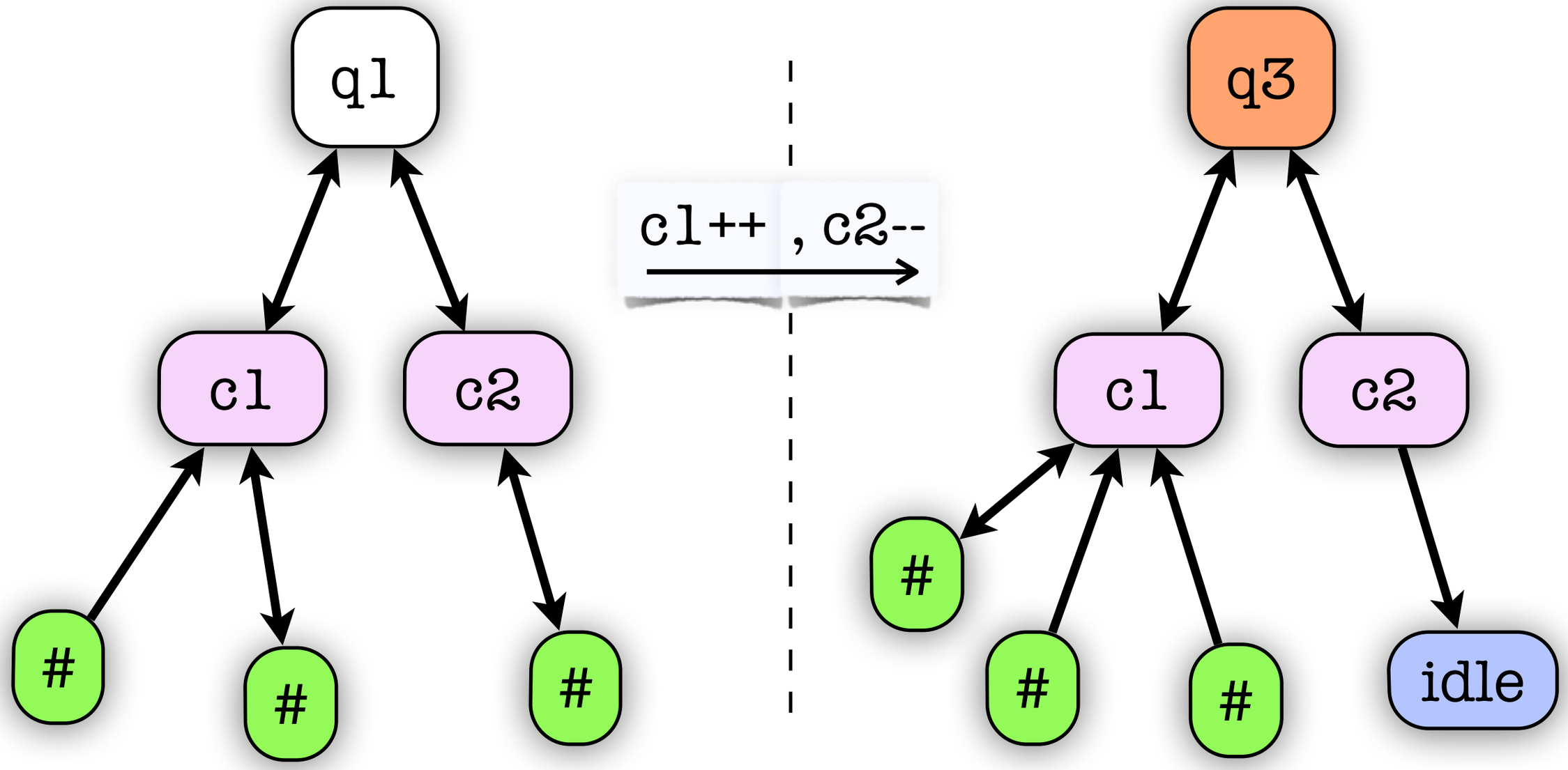


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

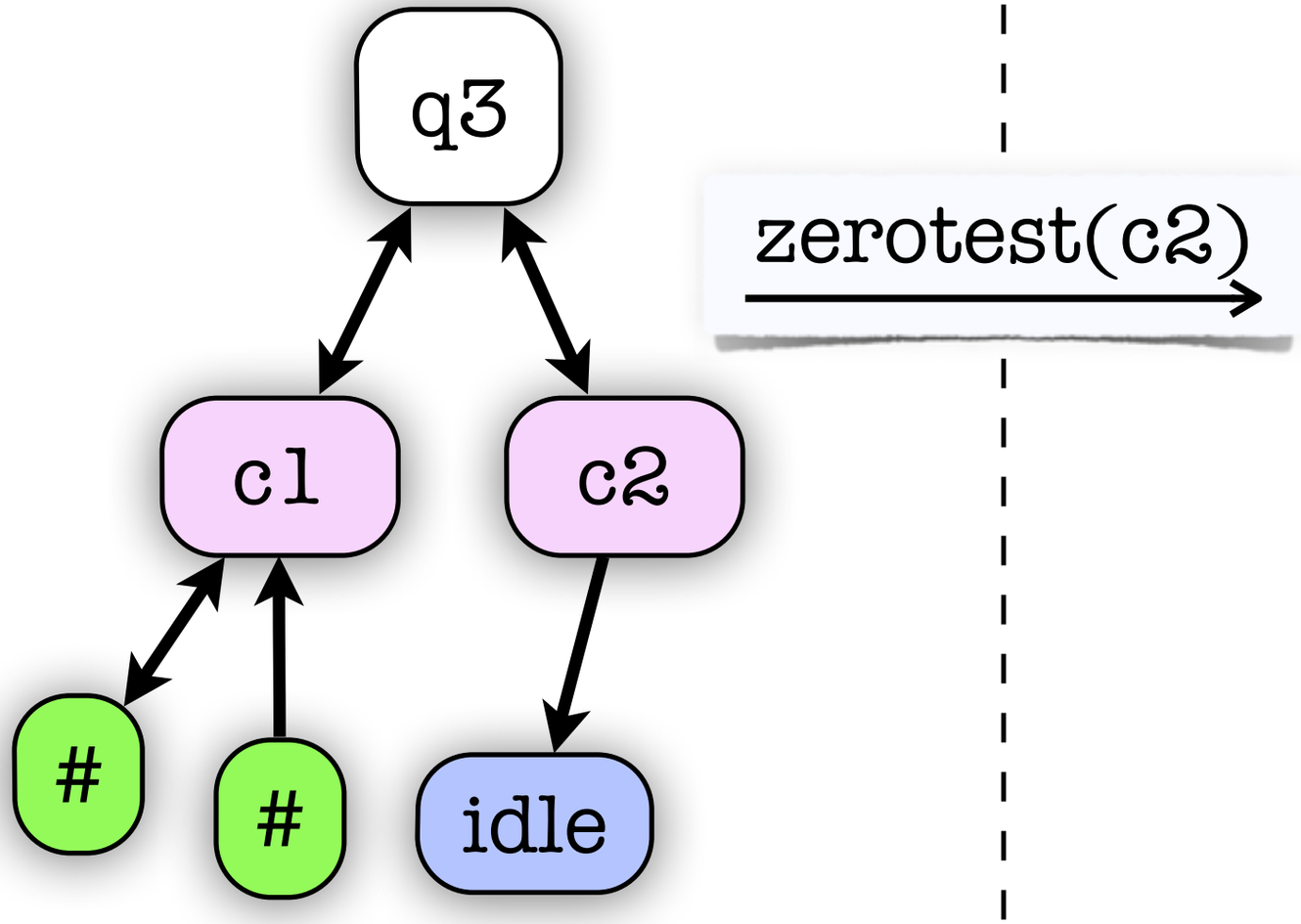


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

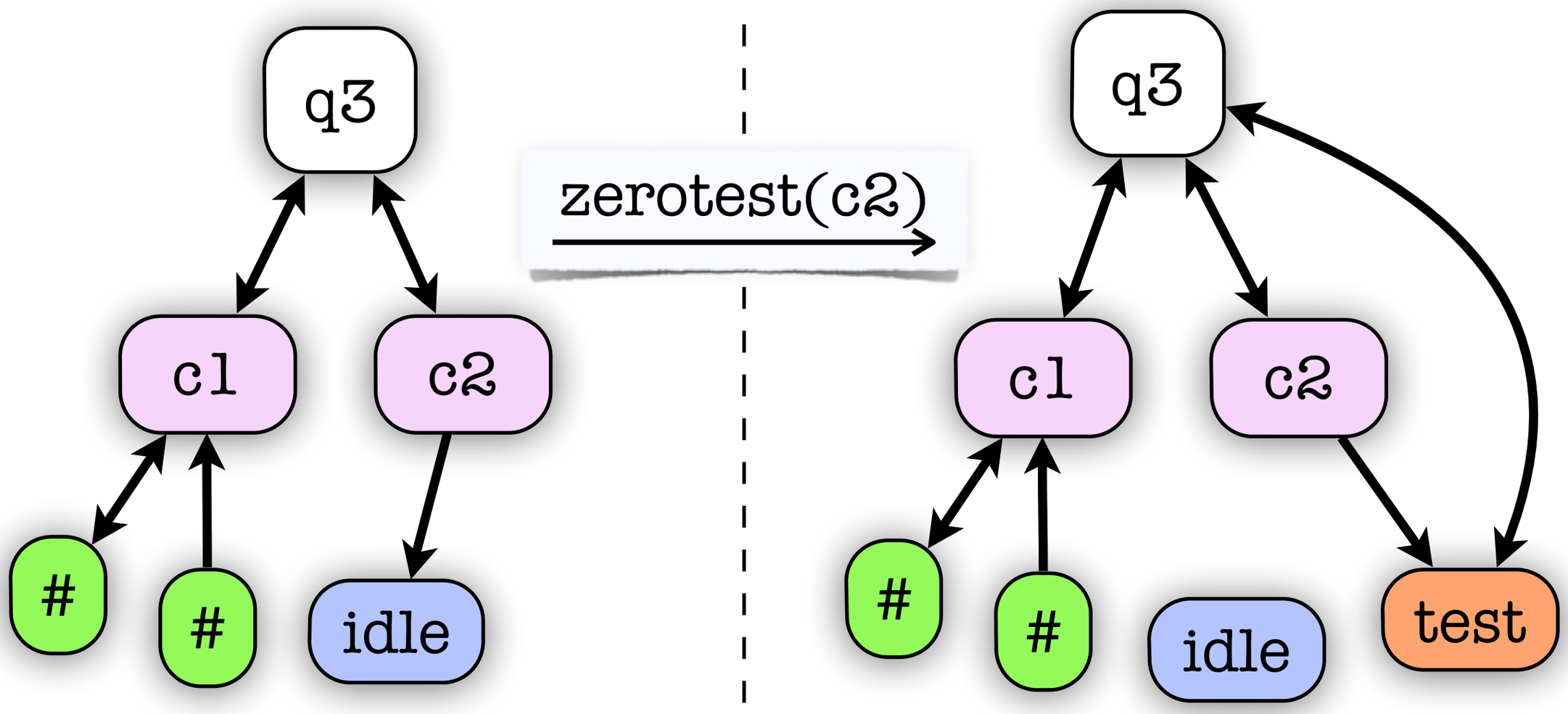


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

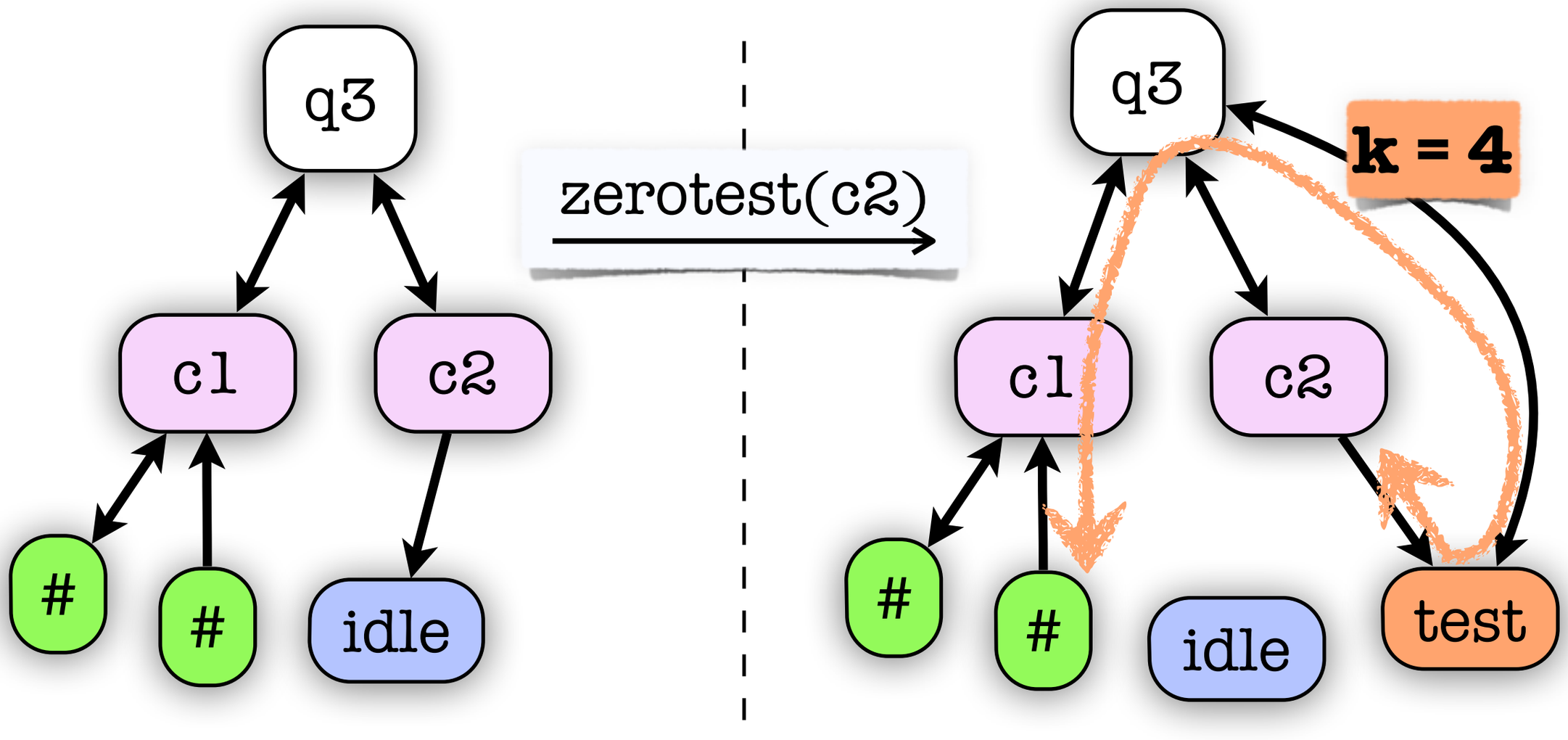
Two counters Minsky Machine to 4-bounded reachability



Proofs

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

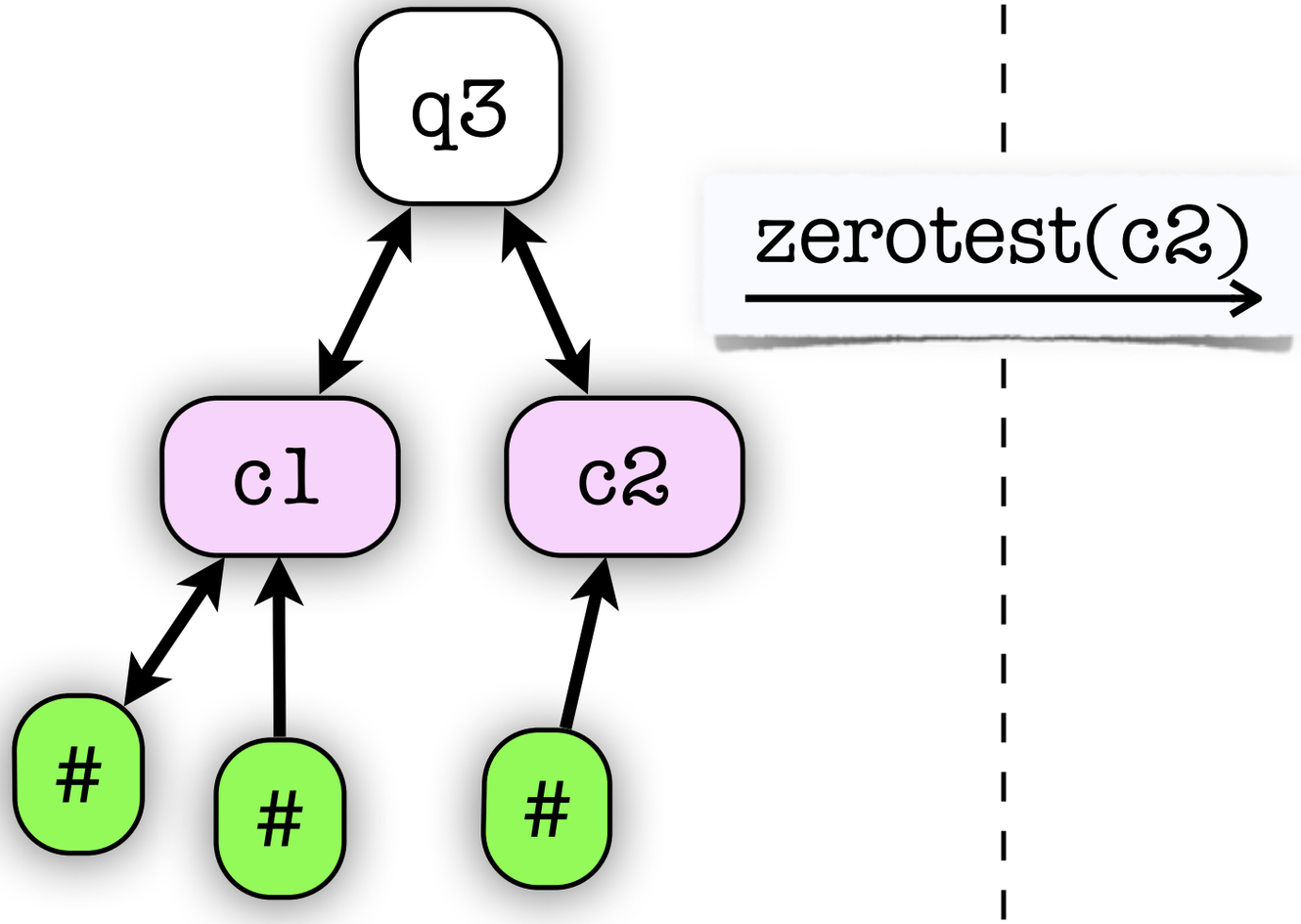


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

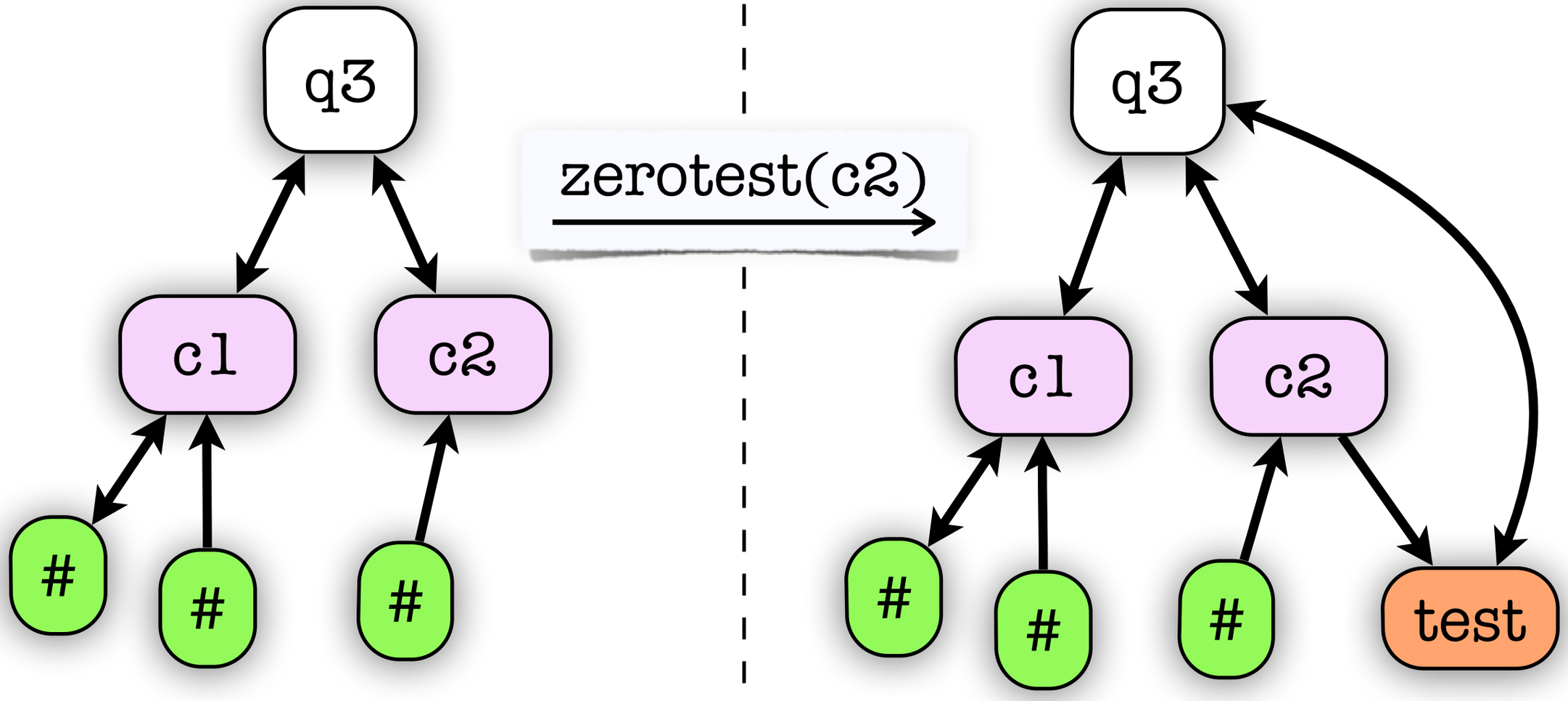


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

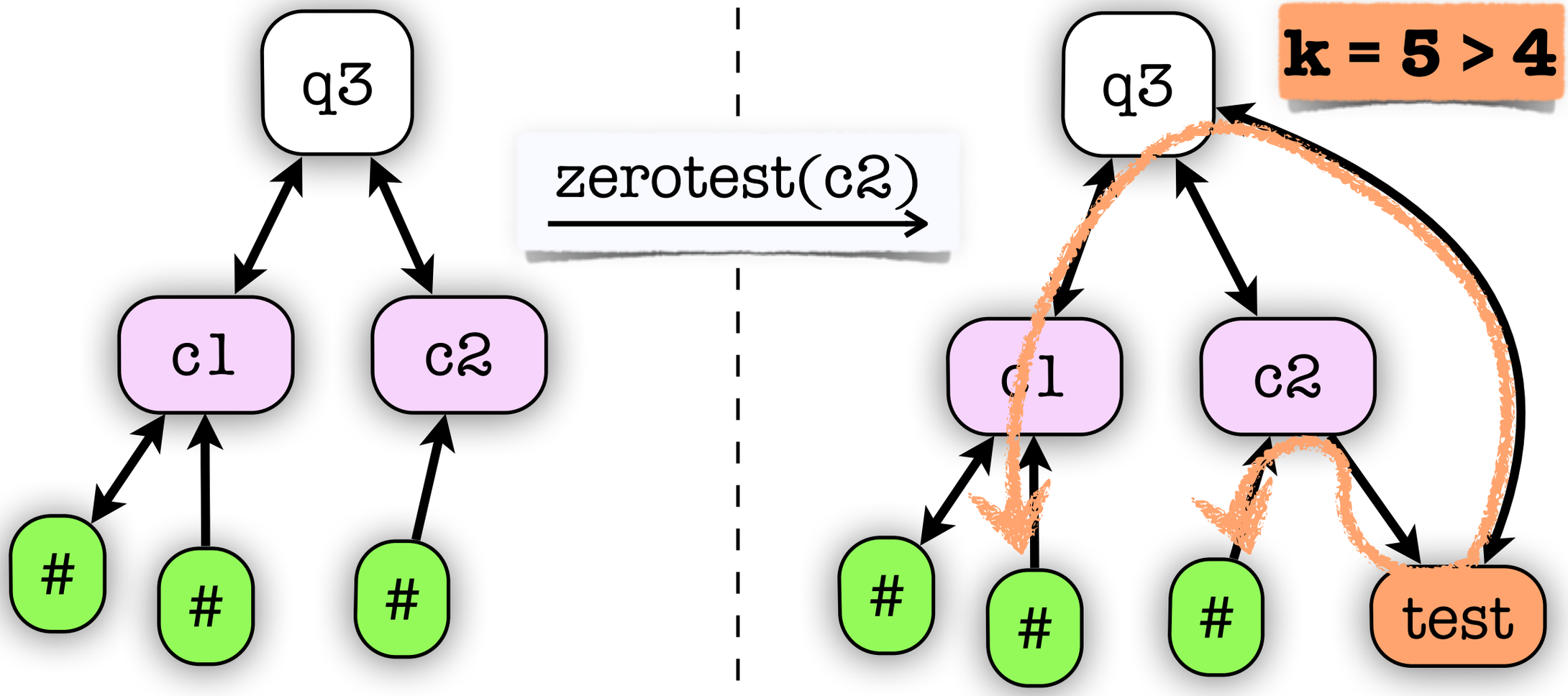


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability

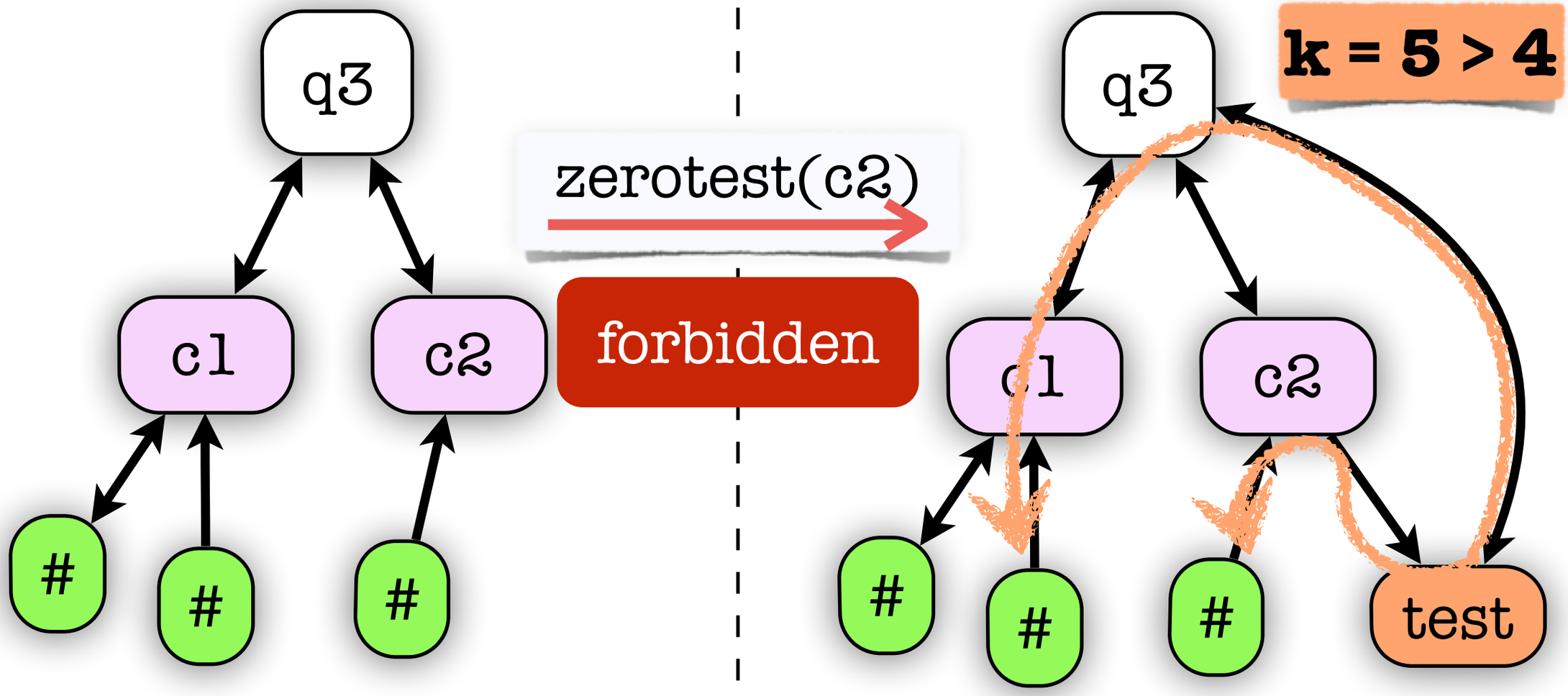


Proofs

Verification of Buffered Dynamic Register Automata

Bound Buffers + Strongly Bounded Simple Path

Two counters Minsky Machine to 4-bounded reachability



Proofs

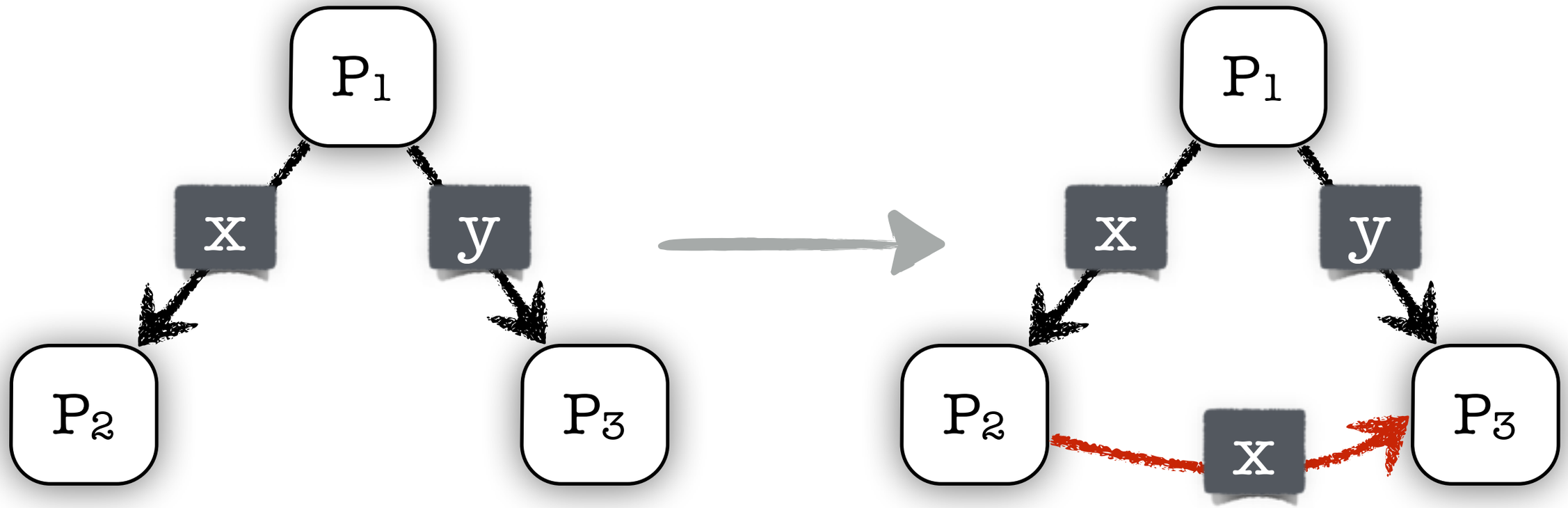
Verification of Buffered Dynamic Register Automata

Acyclic + Bound the Buffer + Strongly Bound Simple Path

Proofs

Verification of Buffered Dynamic Register Automata

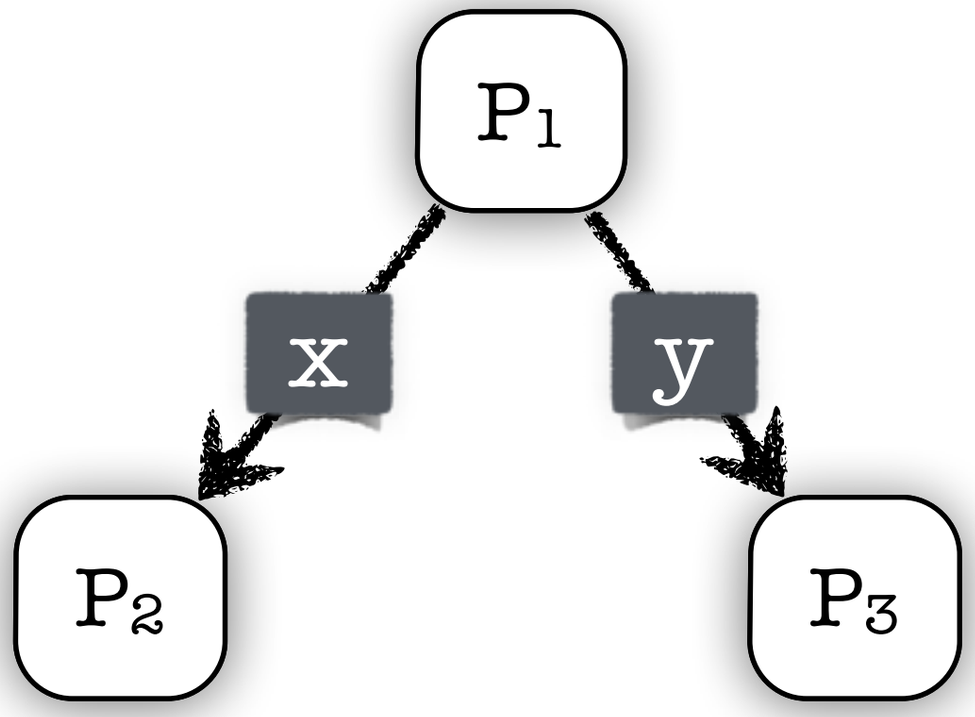
Acyclic + Bound the Buffer + Strongly Bound Simple Path



Messages can **not** be sent with process IDs —
Otherwise **cycles** would be created.

Proofs

Acyclic + Bound the Buffer + Strongly Bound Simple Path

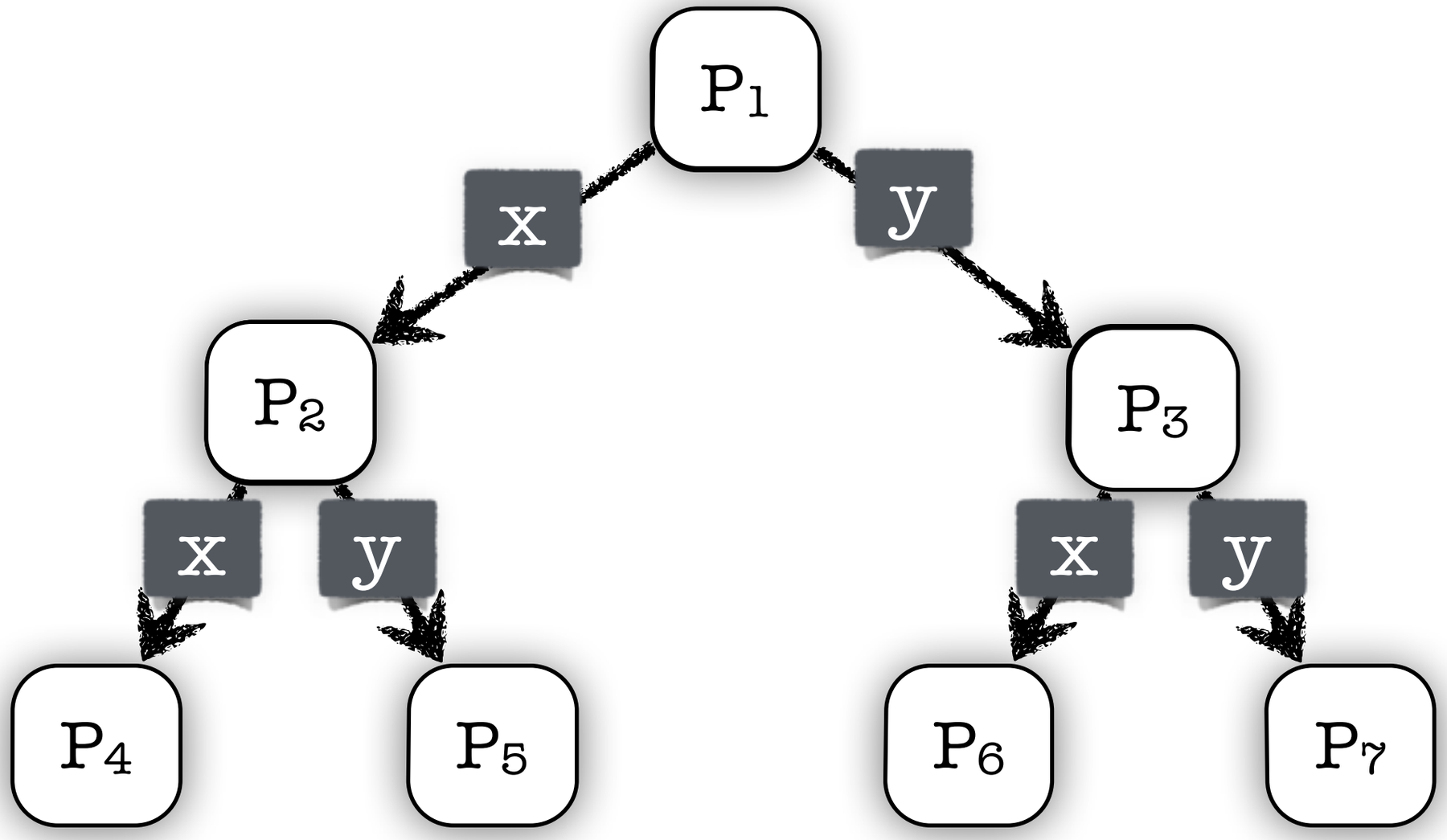


Finite # Registers (Suppose x and y)

Proofs

Verification of Buffered Dynamic Register Automata

Acyclic + Bound the Buffer + Strongly Bound Simple Path

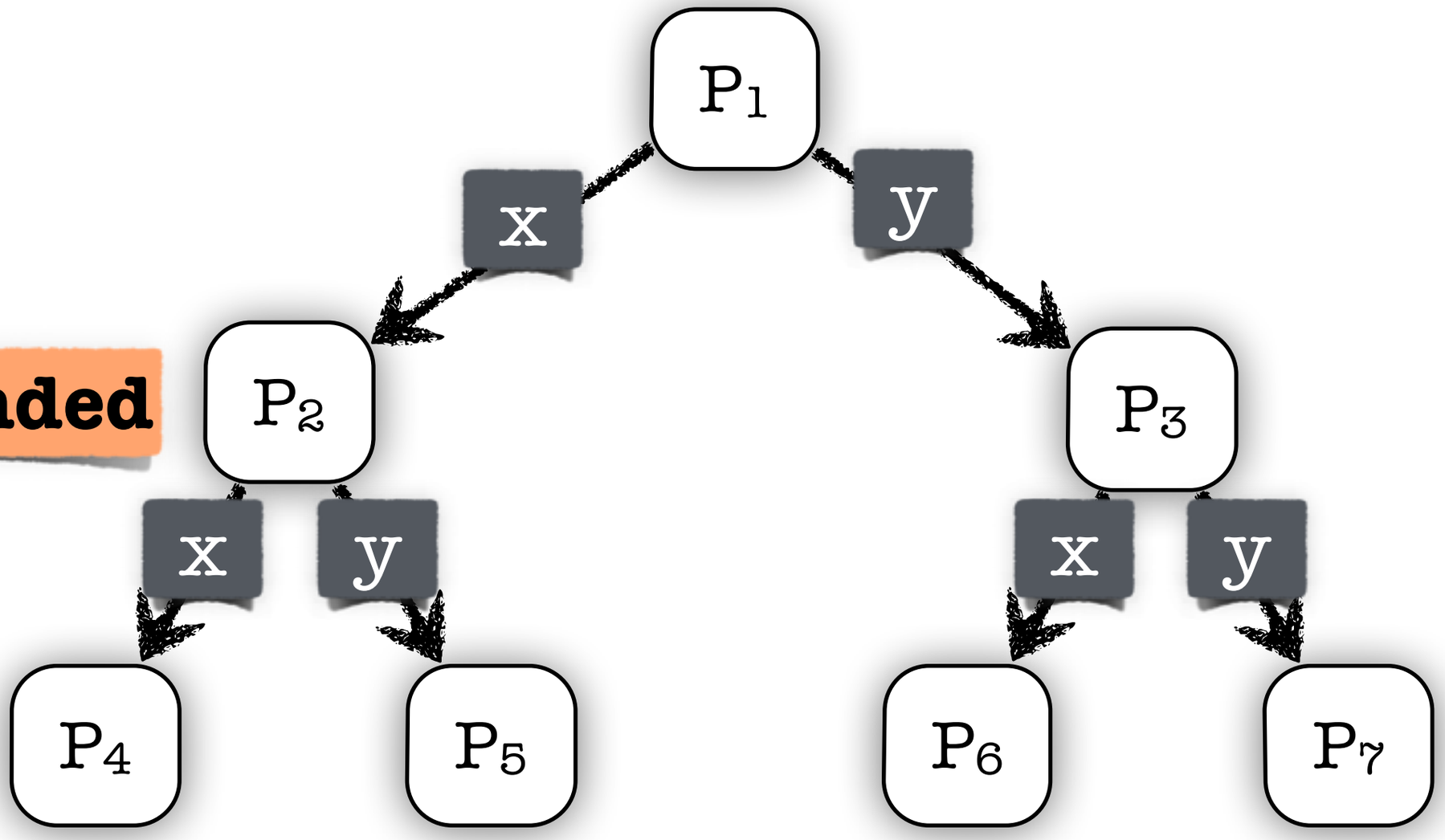


Proofs

Verification of Buffered Dynamic Register Automata

Acyclic + Bound the Buffer + Strongly Bound Simple Path

2-Bounded

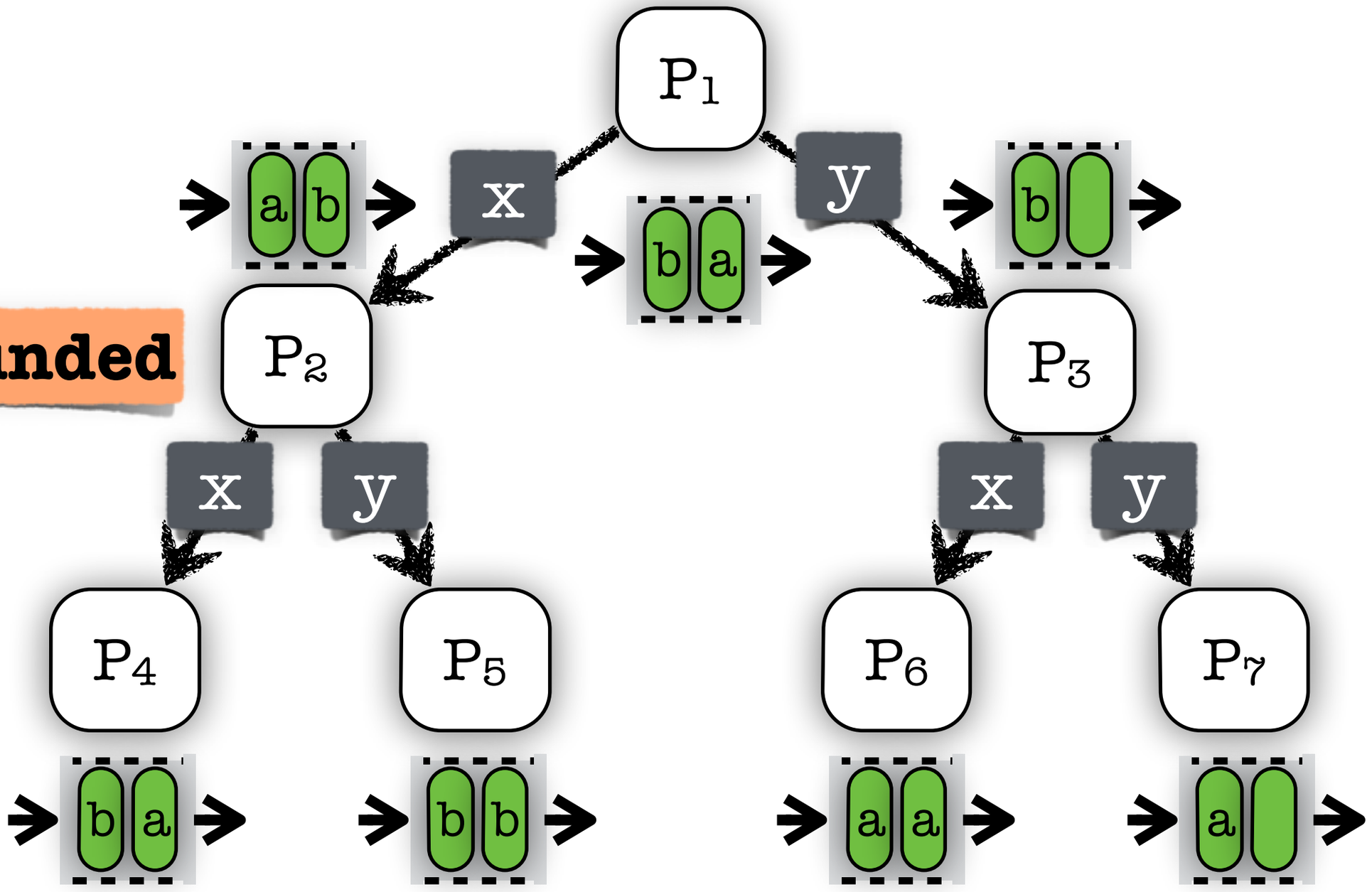


Proofs

Verification of Buffered Dynamic Register Automata

Acyclic + Bound the Buffer + Strongly Bound Simple Path

2-Bounded

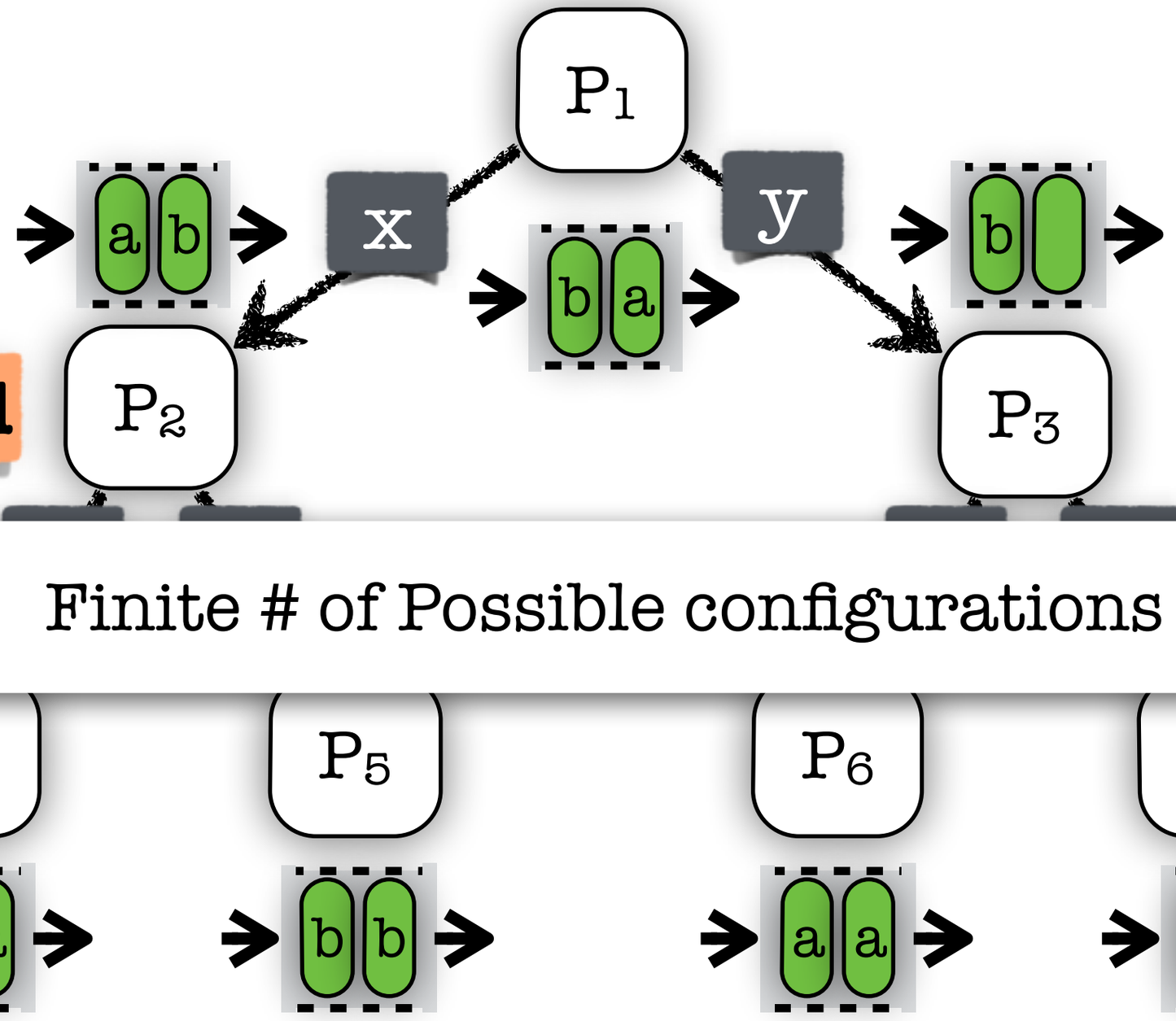


Proofs

Verification of Buffered Dynamic Register Automata

Acyclic + Bound the Buffer + Strongly Bound Simple Path

2-Bounded



Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Well-Structured Transition Systems

[Abdulla et al. 1996], [Finkel et al. 2001]

Symbolic representation of infinite set of configurations

Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations
- Prove **Monotonicity** of Transition Relation
- Provide an algorithm to compute the **Pre** of an upward closed set

Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations
- Prove **Monotonicity** of Transition Relation
- Provide an algorithm to compute the **Pre** of an upward closed set

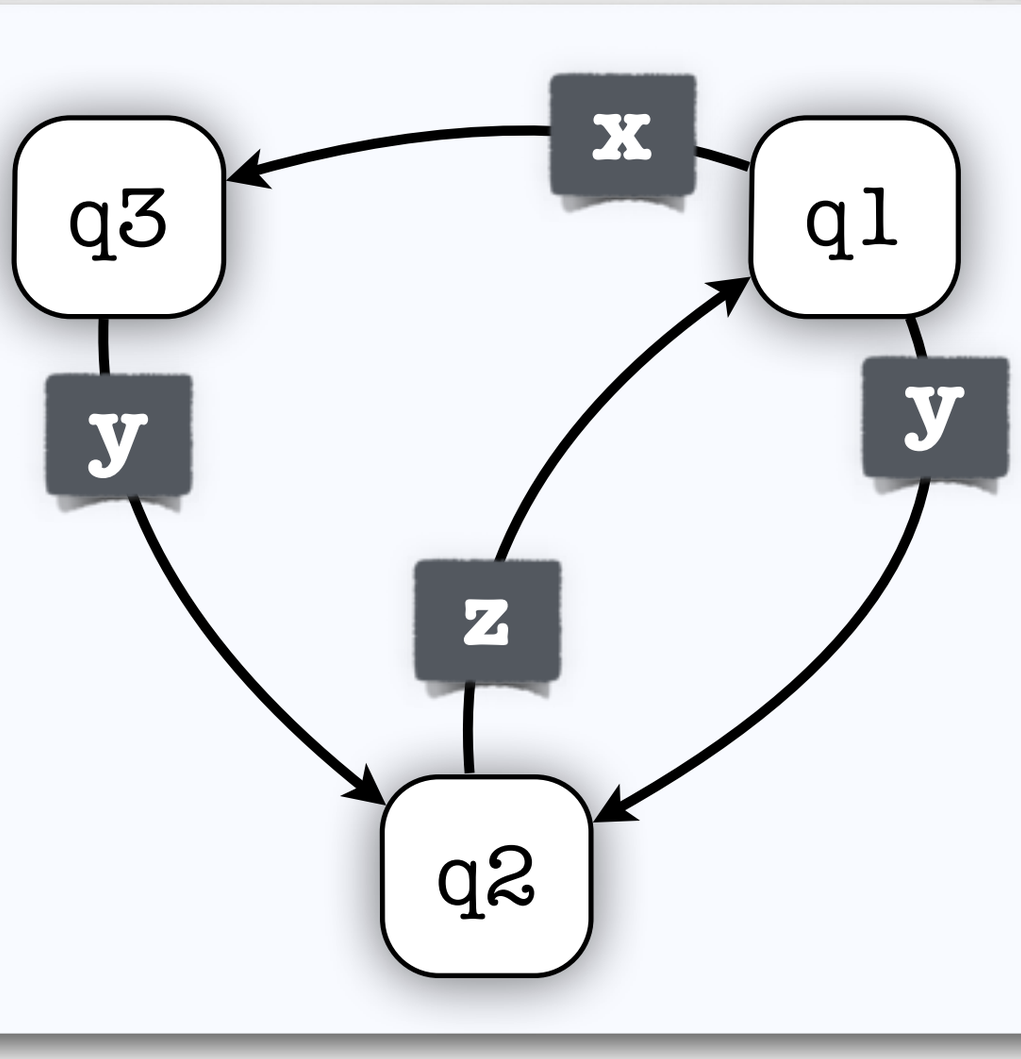
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations

Ordering: subgraph relation



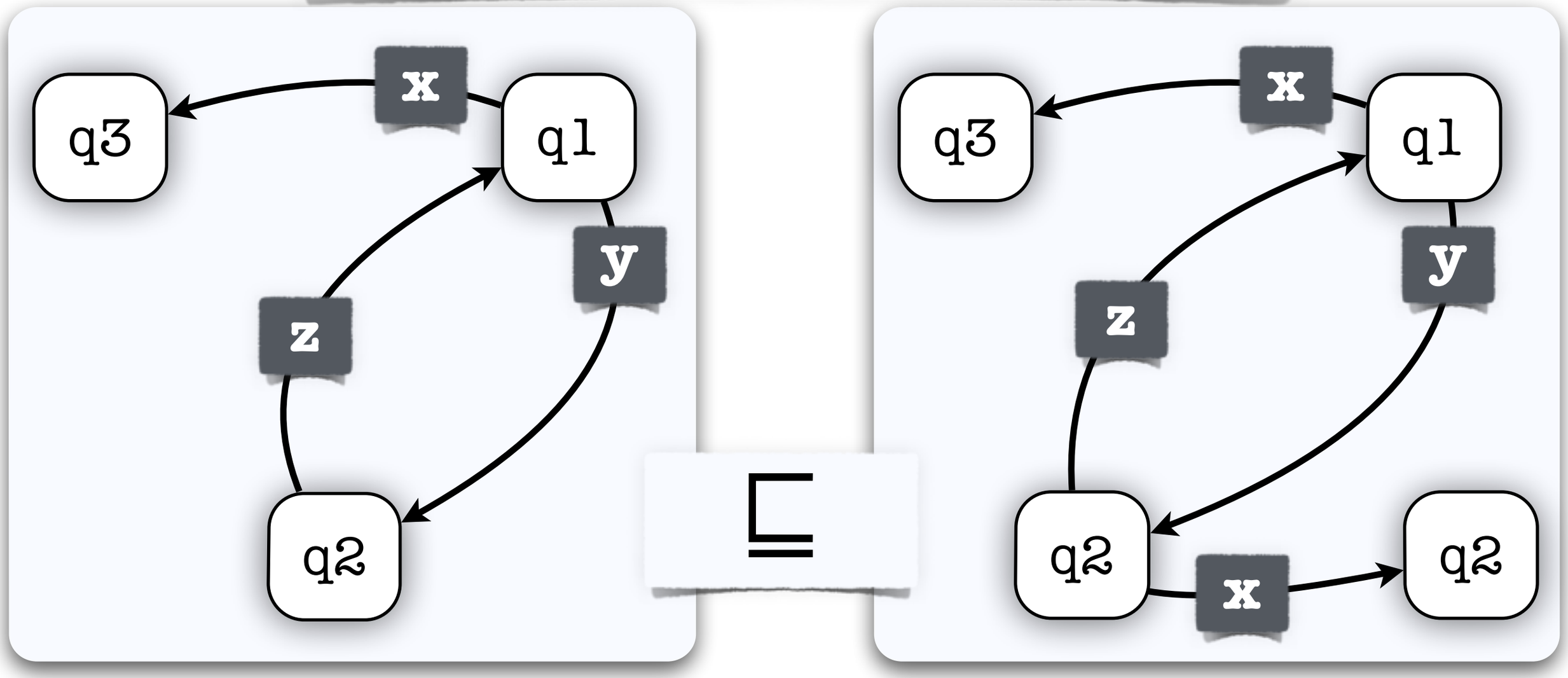
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations

Ordering: induced subgraph relation



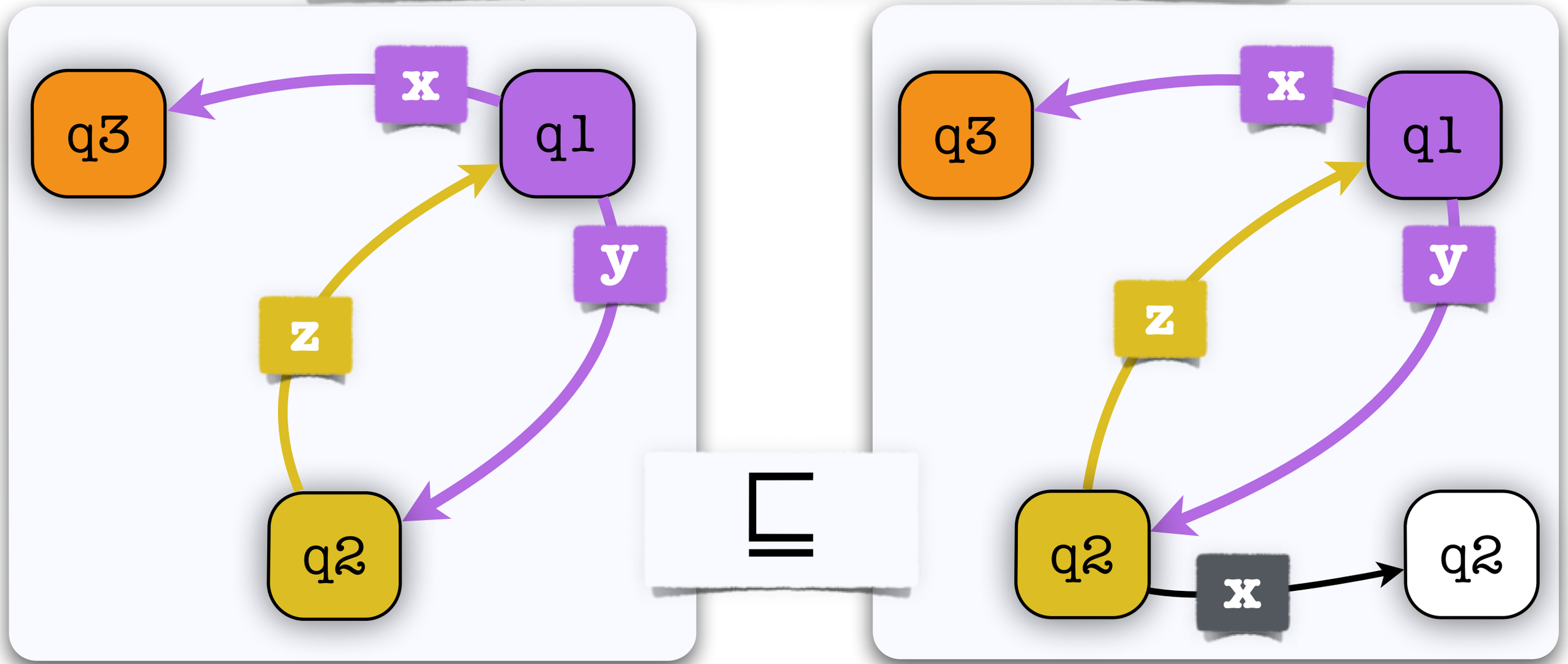
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations

Ordering: induced subgraph relation



Proofs

Well-Structured Transition Systems

Lo… …mple Path

- Define a **Well-Quasi Order** on configurations

Ordering: induced subgraph relation

[Ding, 1992] Subgraphs and Well-Quasi Ordering

(Induced) subgraph relation is a WQO on Strongly-Bounded graphs

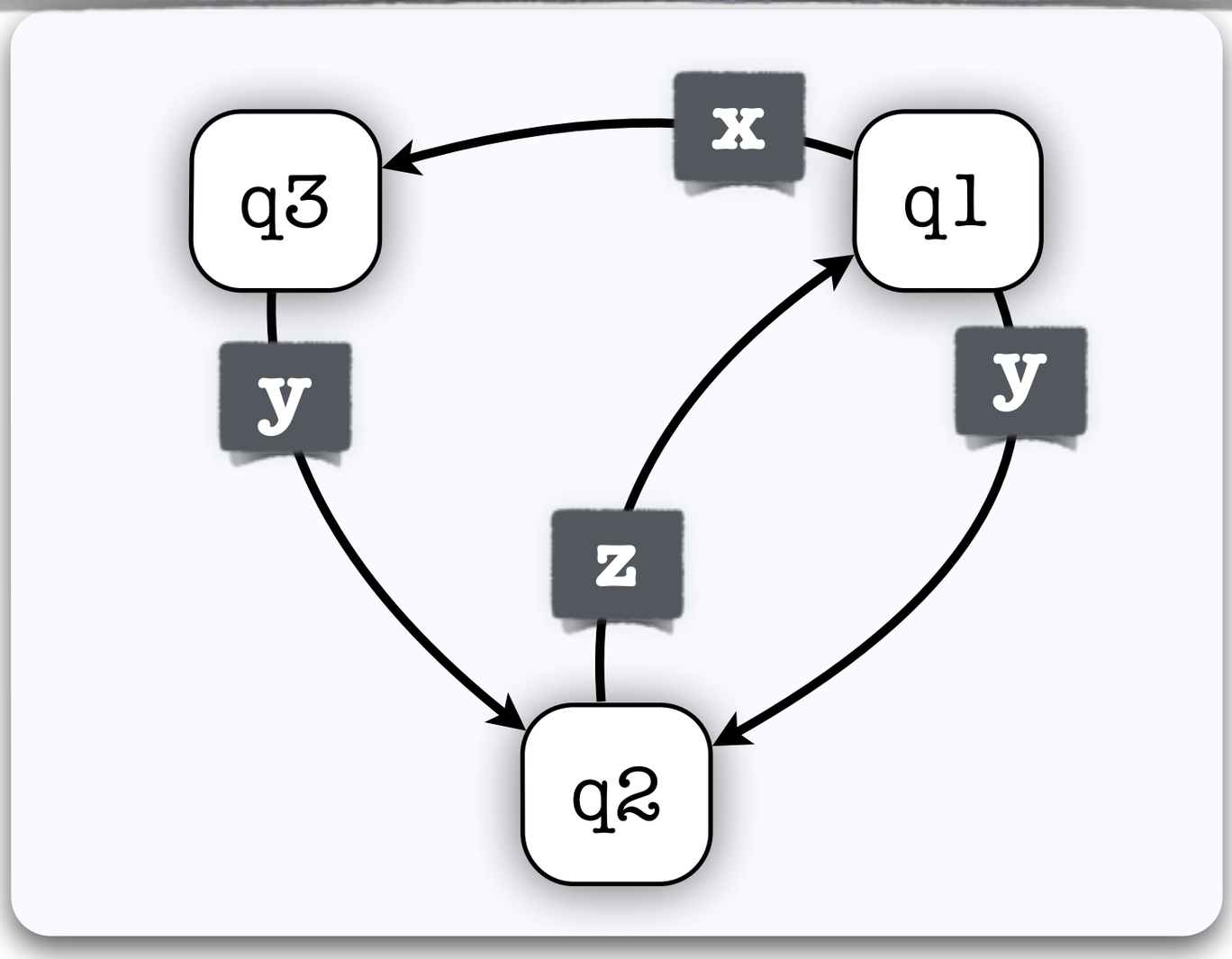
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations

Ordering: induced subgraph relation



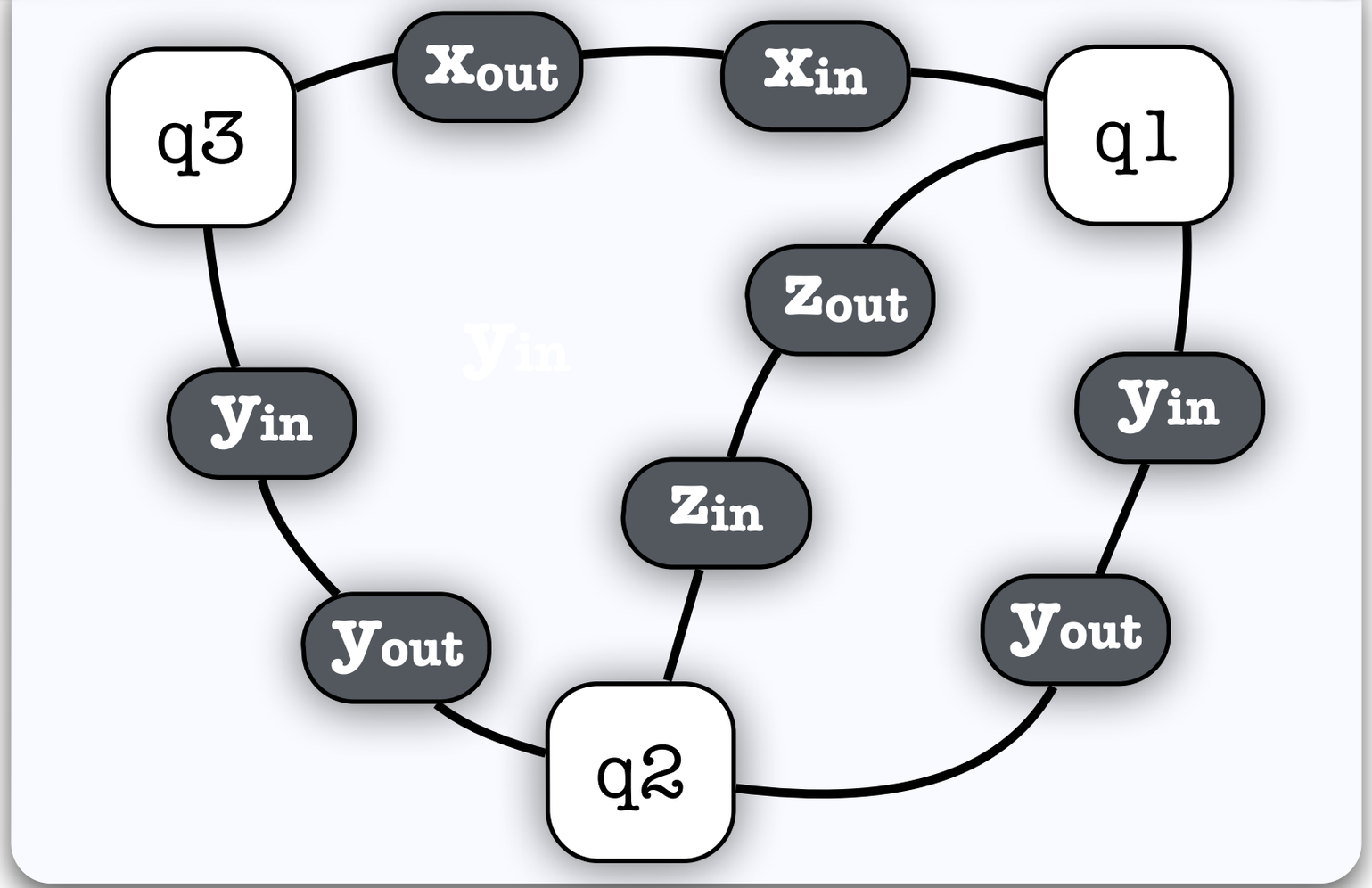
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations

Ordering: induced subgraph relation



Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations
- Prove **Monotonicity** of Transition Relation
- Provide an algorithm to compute the **Pre** of an upward closed set

Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations
- Prove **Monotonicity** of Transition Relation
- Provide an algorithm to compute the **Pre** of an upward closed set

Strongly Safe DRA

Well-Structured Transition Systems

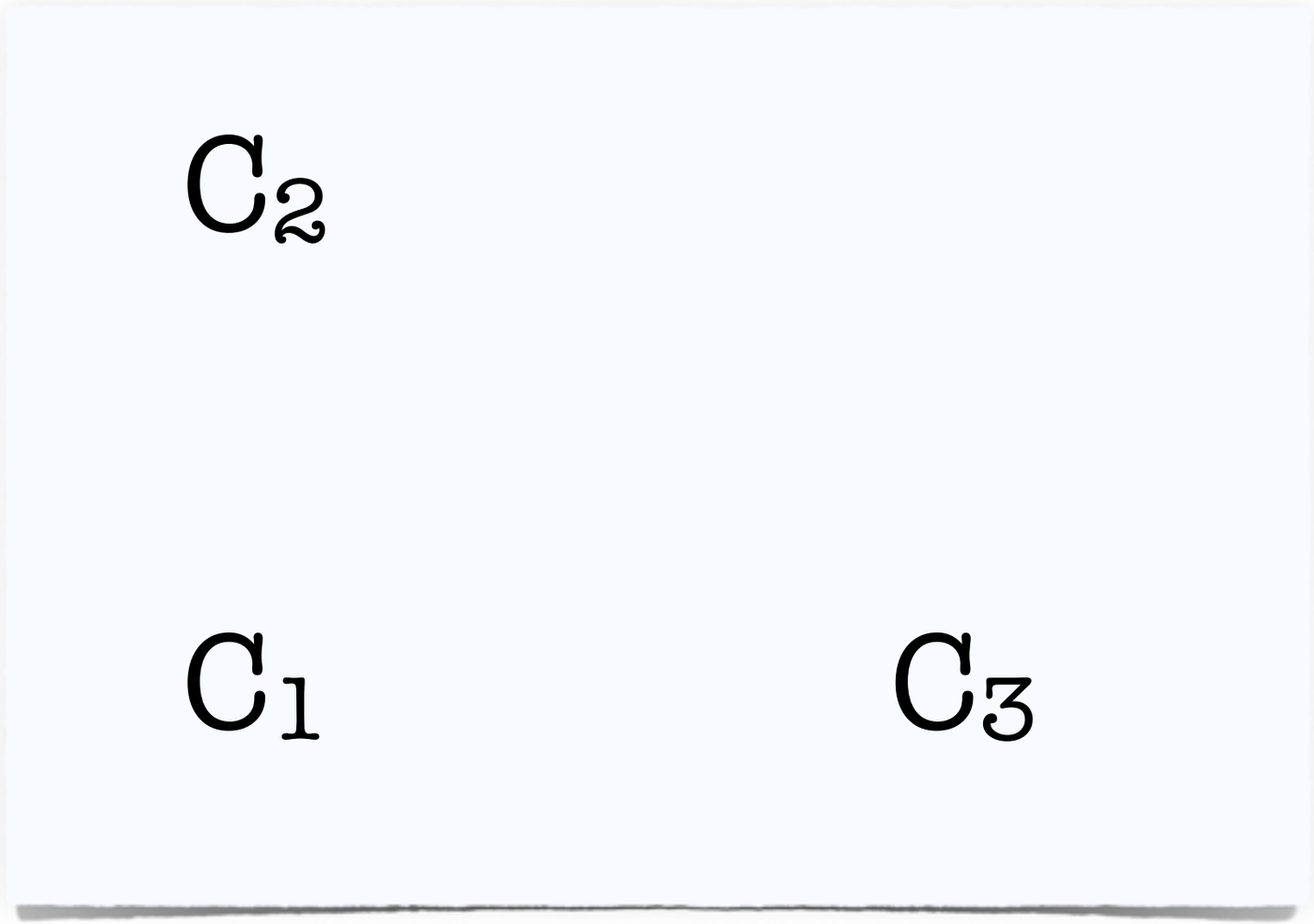
- Define a **Well-Quasi Order** on configurations
- Prove **Monotonicity** of Transition Relation
- Provide an algorithm to compute the **Pre** of an upward closed set

Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Prove **Monotonicity** of Transition Relation



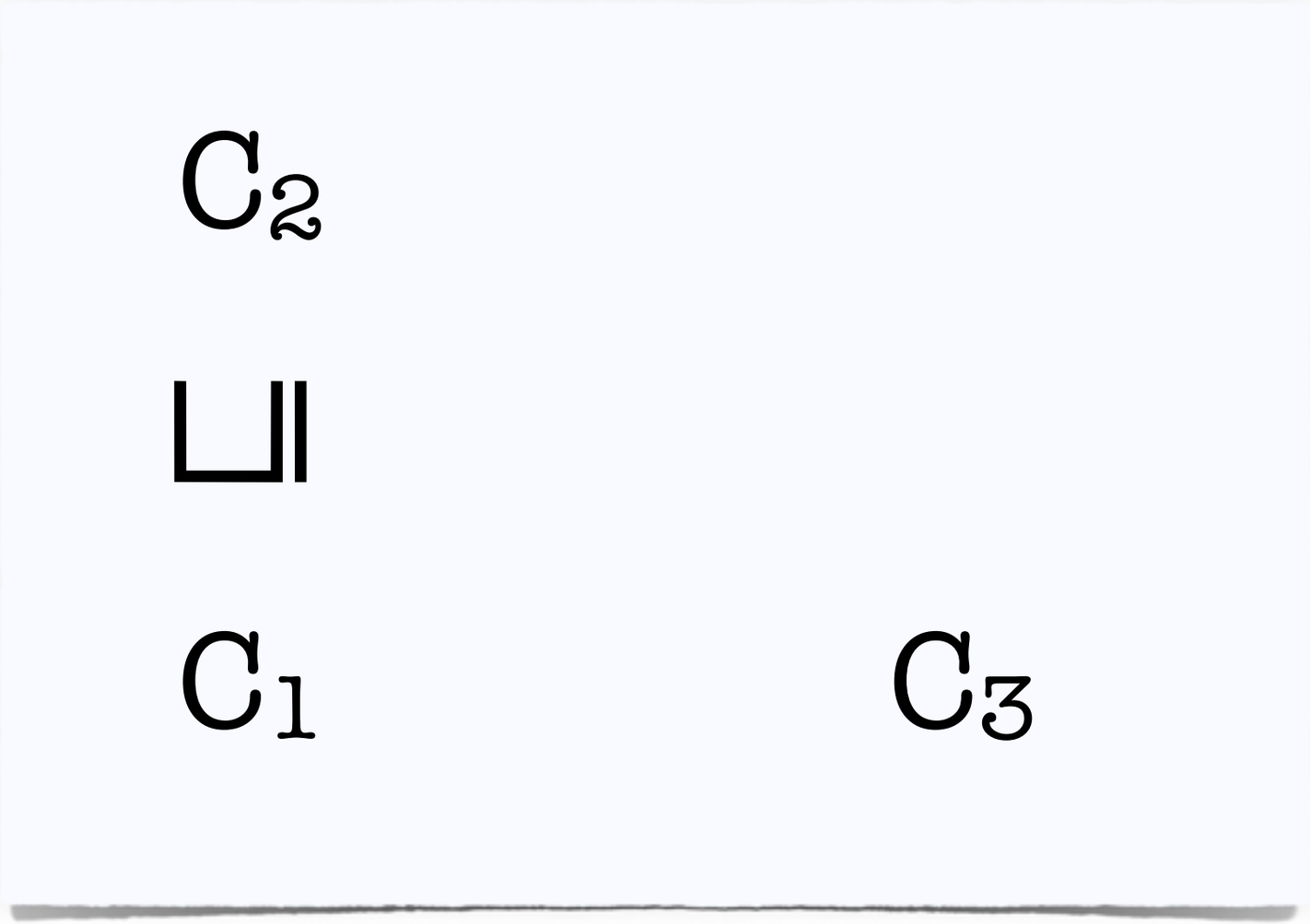
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

Lo… …mple Path

- Prove **Monotonicity** of Transition Relation



Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

Lo… …mple Path

- Prove **Monotonicity** of Transition Relation

C_2

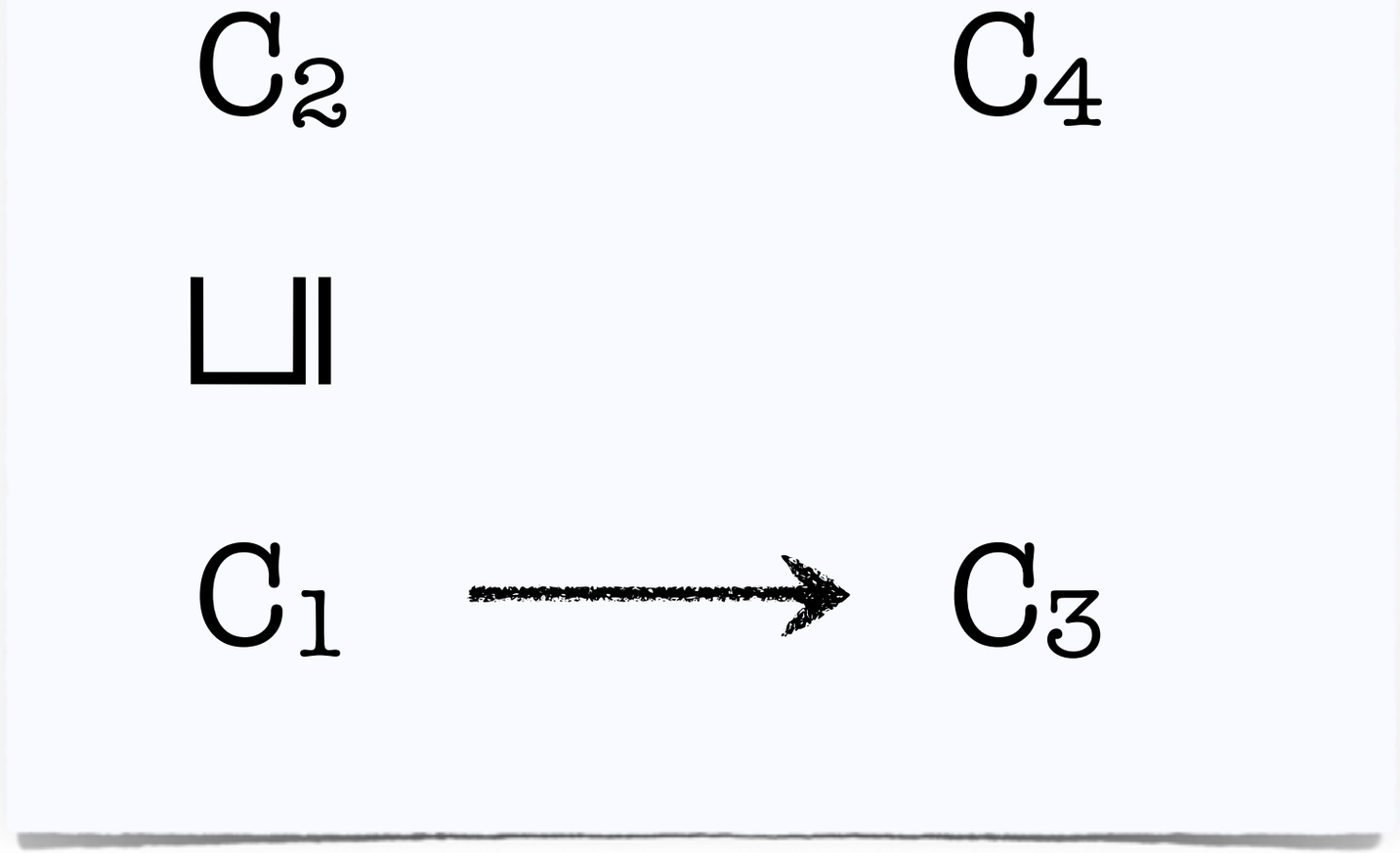
$\sqcup\!|$

$C_1 \longrightarrow C_3$

Proofs

Well-Structured Transition Systems

- Prove **Monotonicity** of Transition Relation



Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

Lo… …mple Path

- Prove **Monotonicity** of Transition Relation

$C_2 \longrightarrow C_4$

$\sqsubseteq \quad \sqsubseteq$

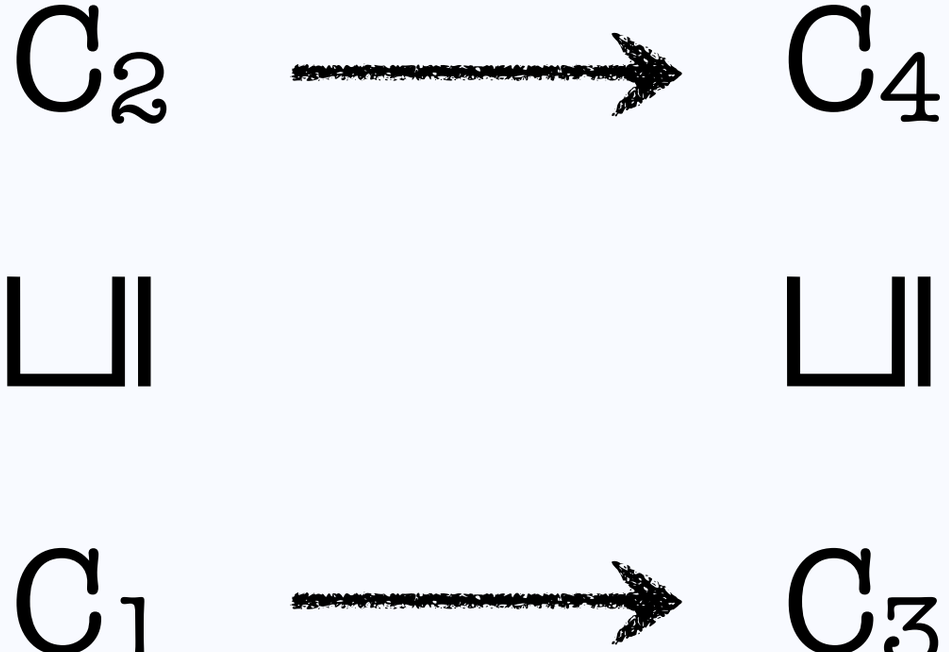
$C_1 \longrightarrow C_3$

Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Prove **Monotonicity** of Transition Relation



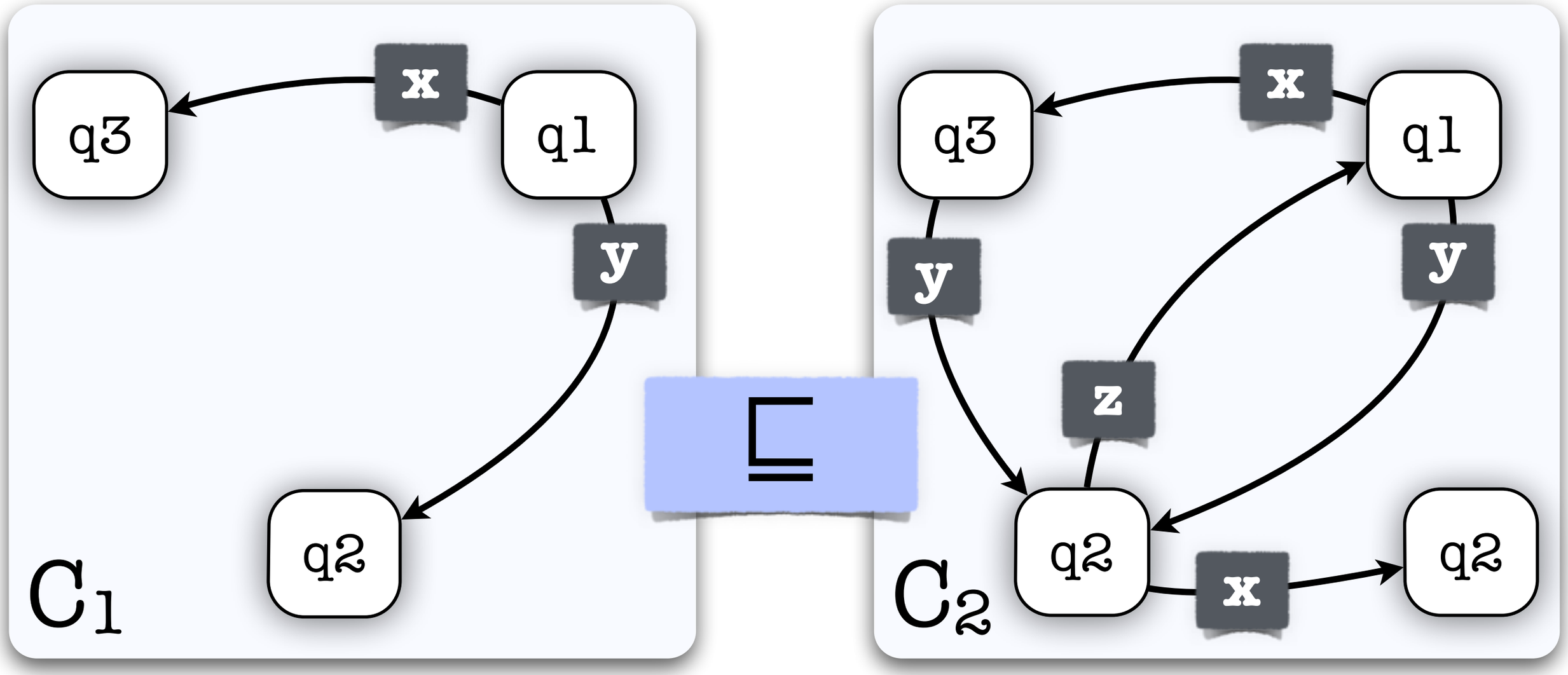
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

Lo… …mple Path

- Prove **Monotonicity** of Transition Relation



Proofs

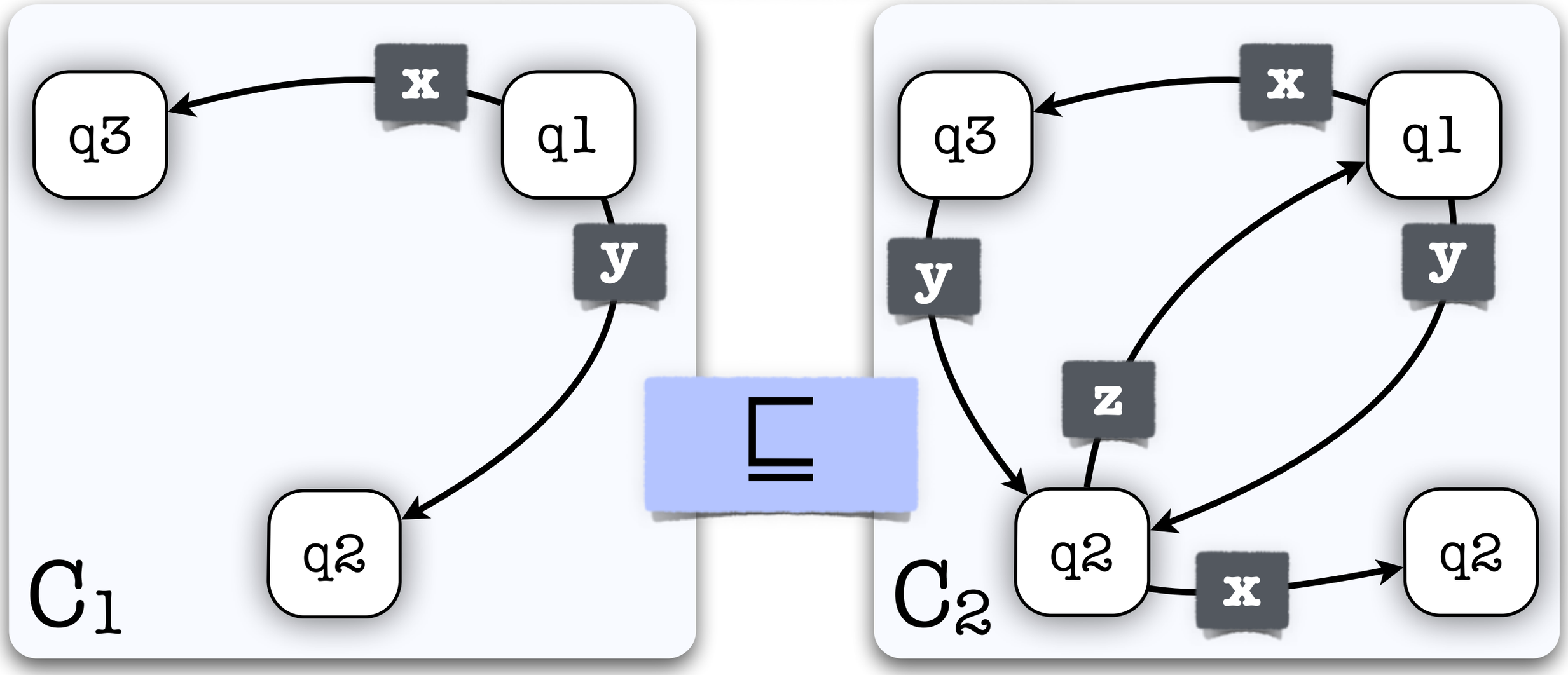
Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

Lo… …mple Path

- Prove **Monotonicity** of Transition Relation

Lossy



Proofs

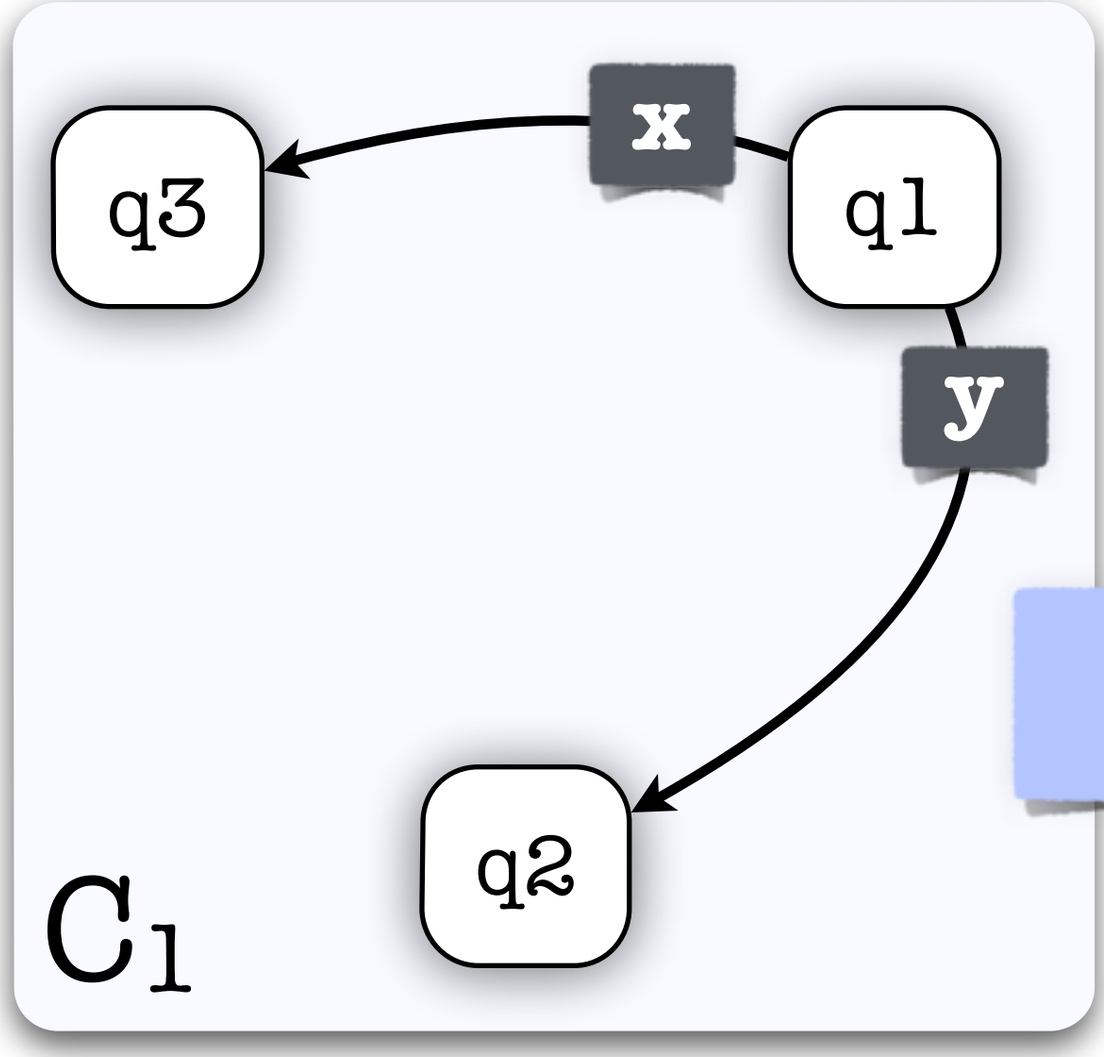
Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

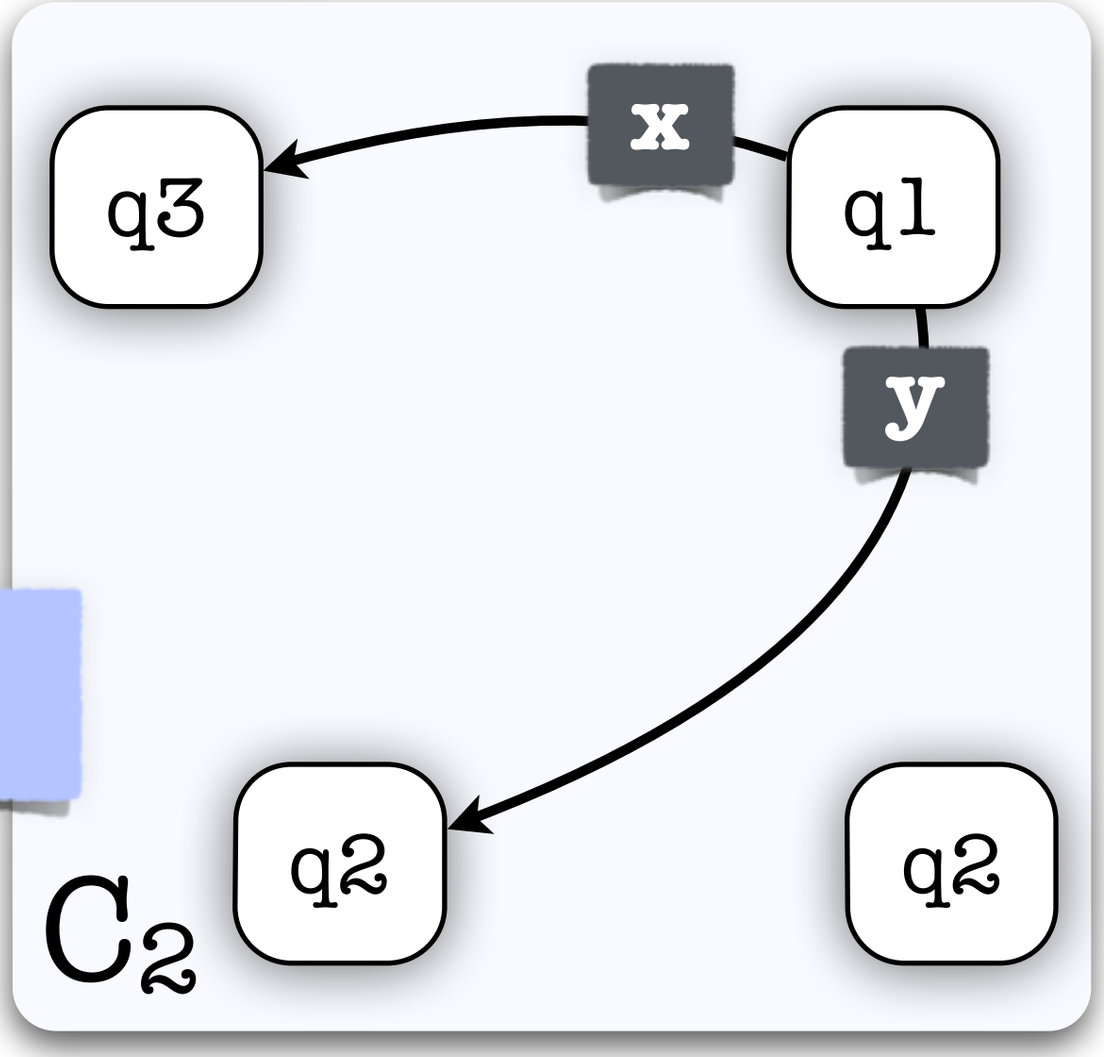
Lo… …mple Path

- Prove **Monotonicity** of Transition Relation

Lossy



⊑



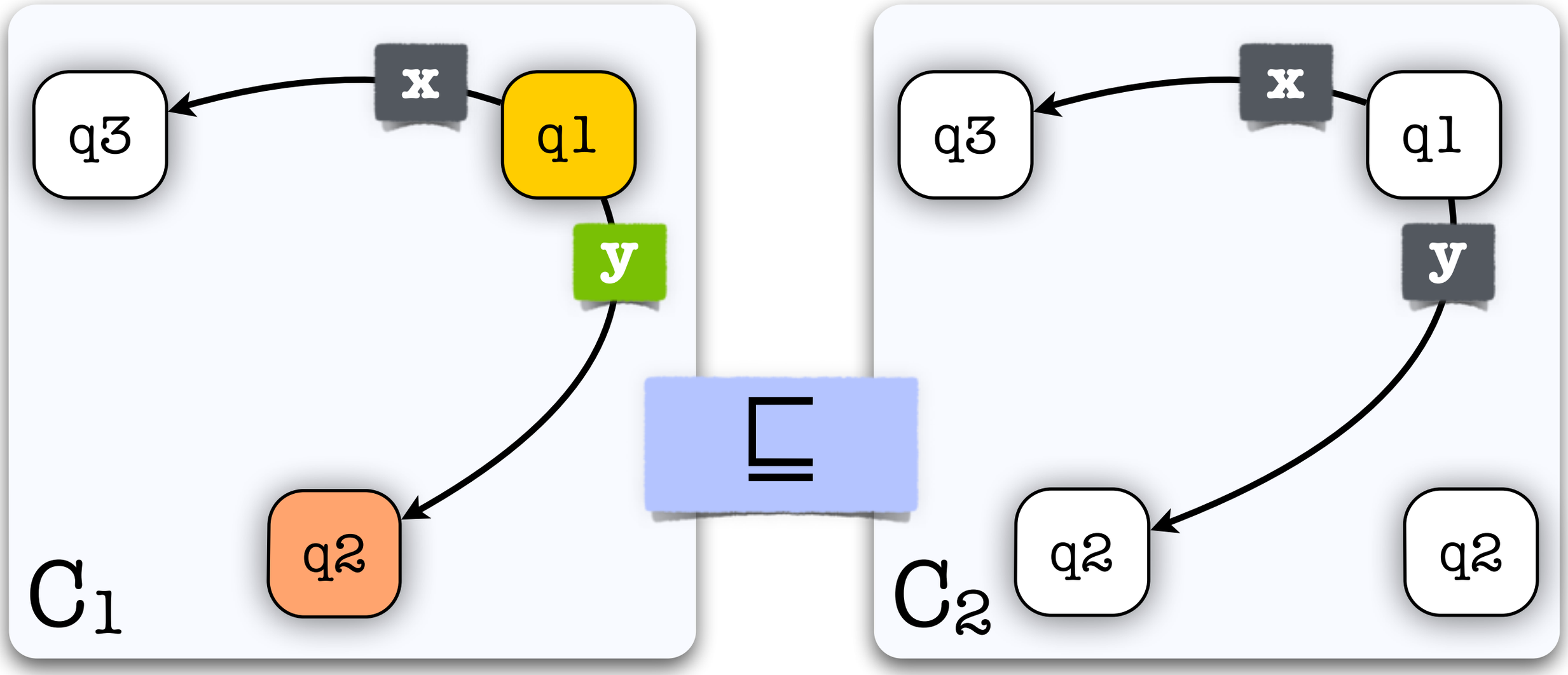
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

Lo… …mple Path

- Prove **Monotonicity** of Transition Relation

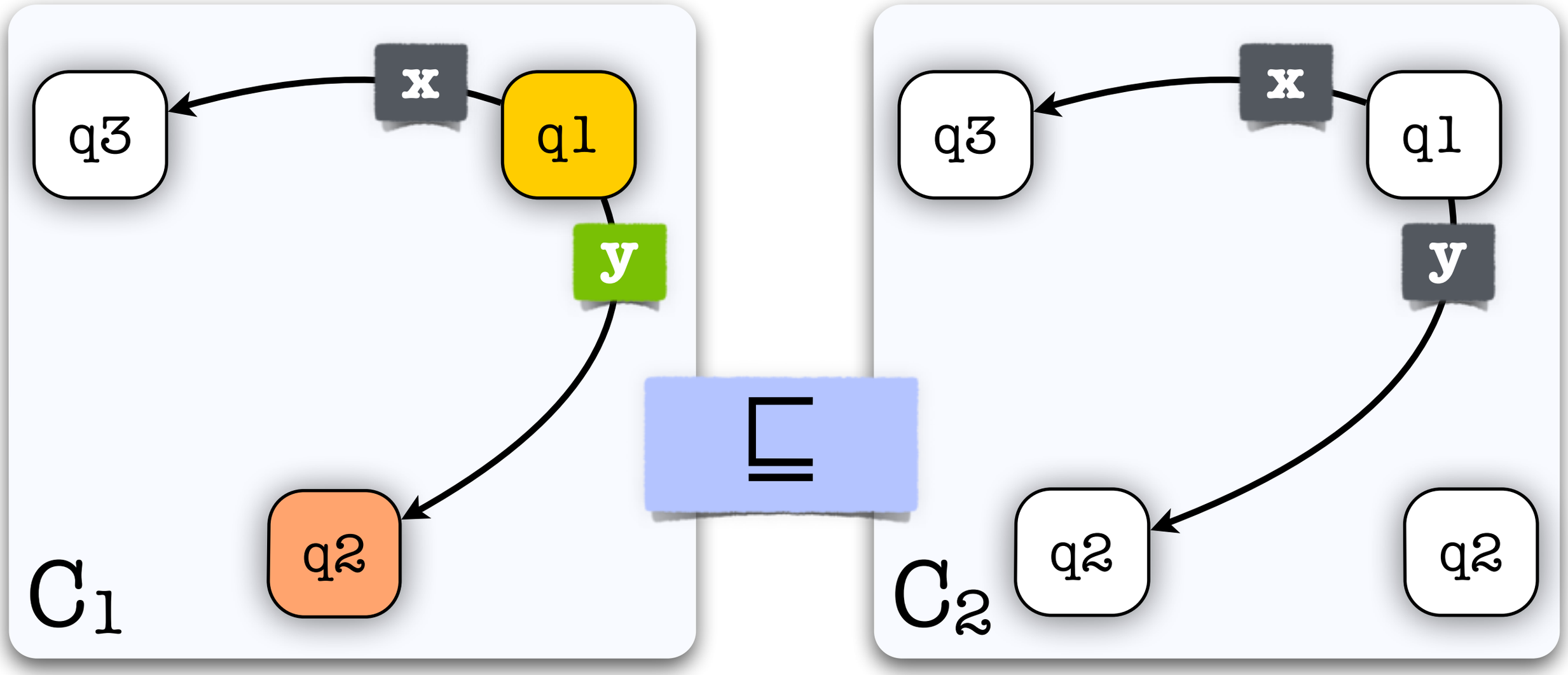


Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Prove **Monotonicity** of Transition Relation



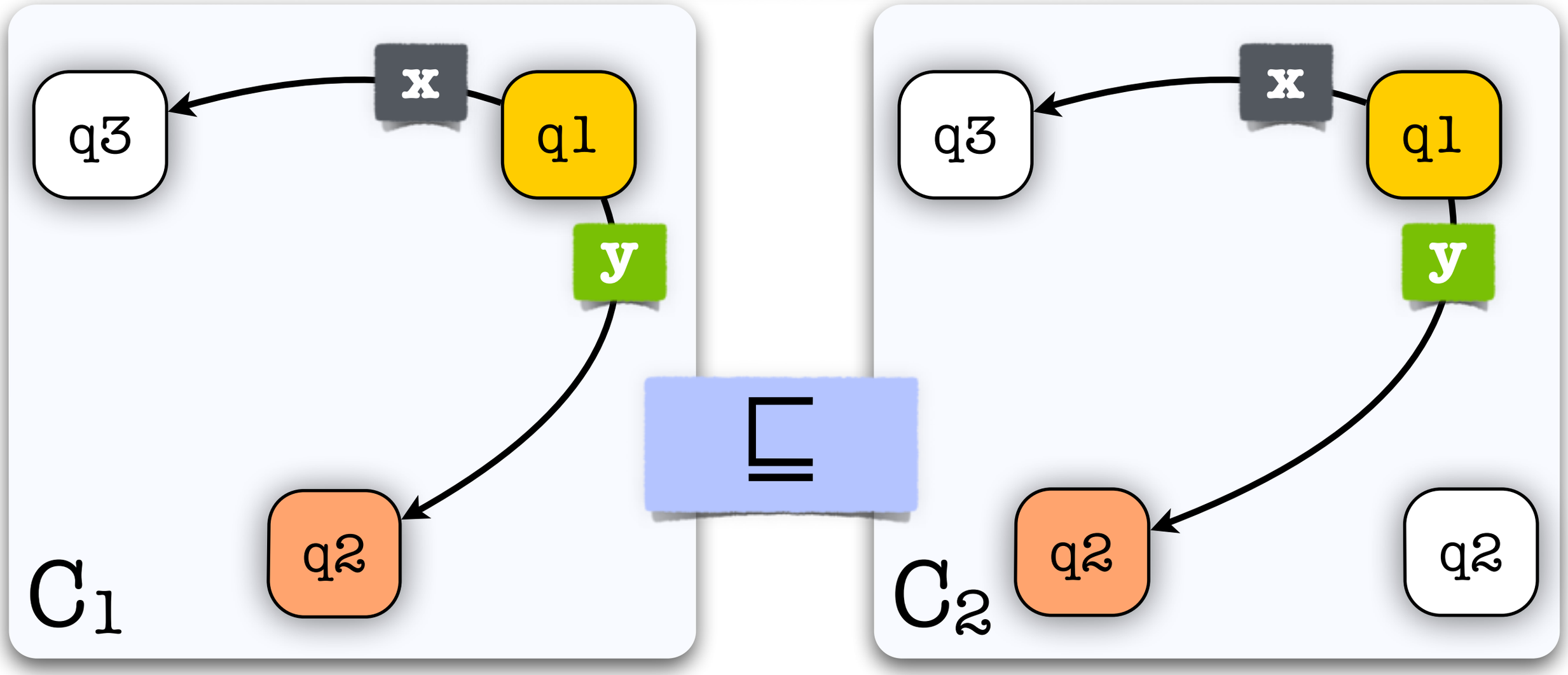
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Prove **Monotonicity** of Transition Relation

Subgraph relation: Register Mapping & States preserved



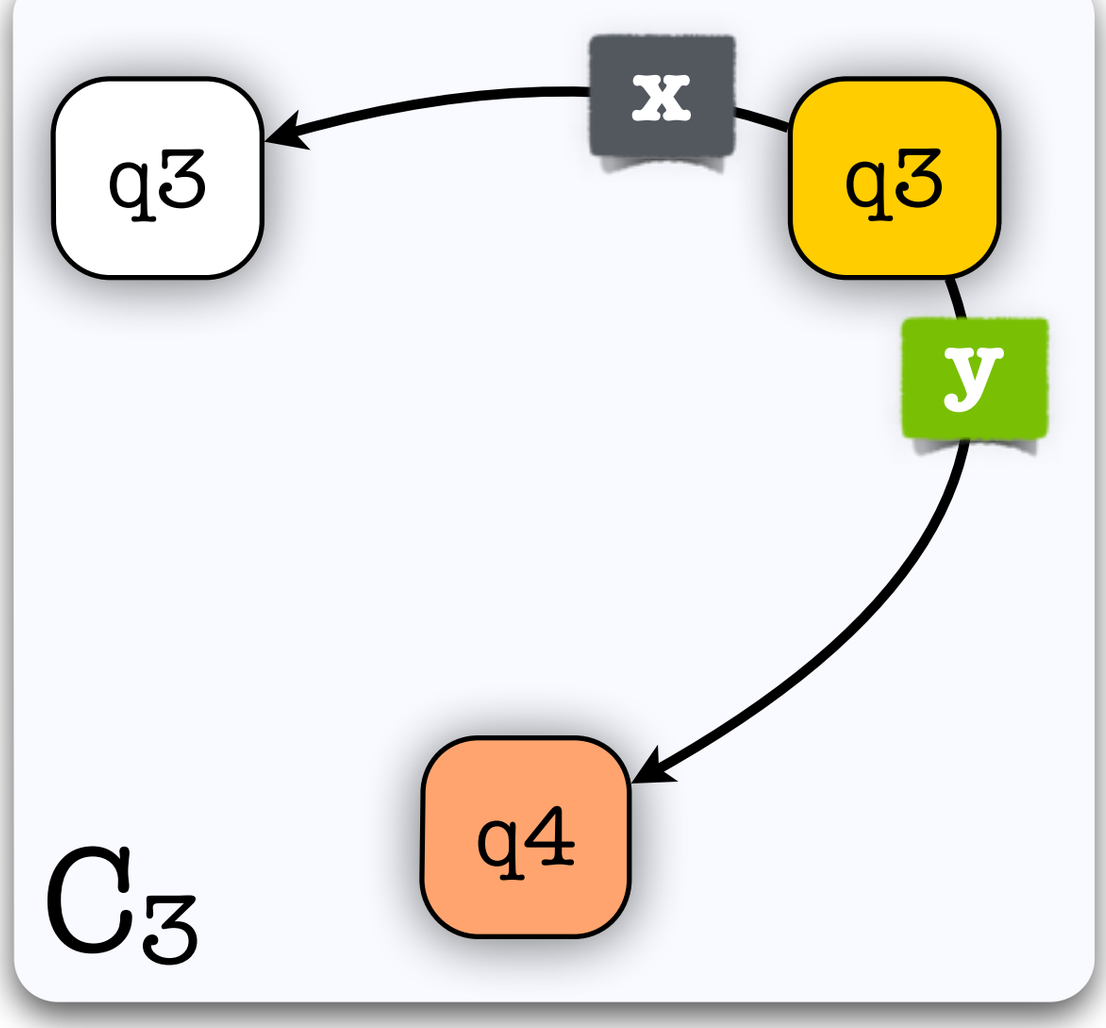
Proofs

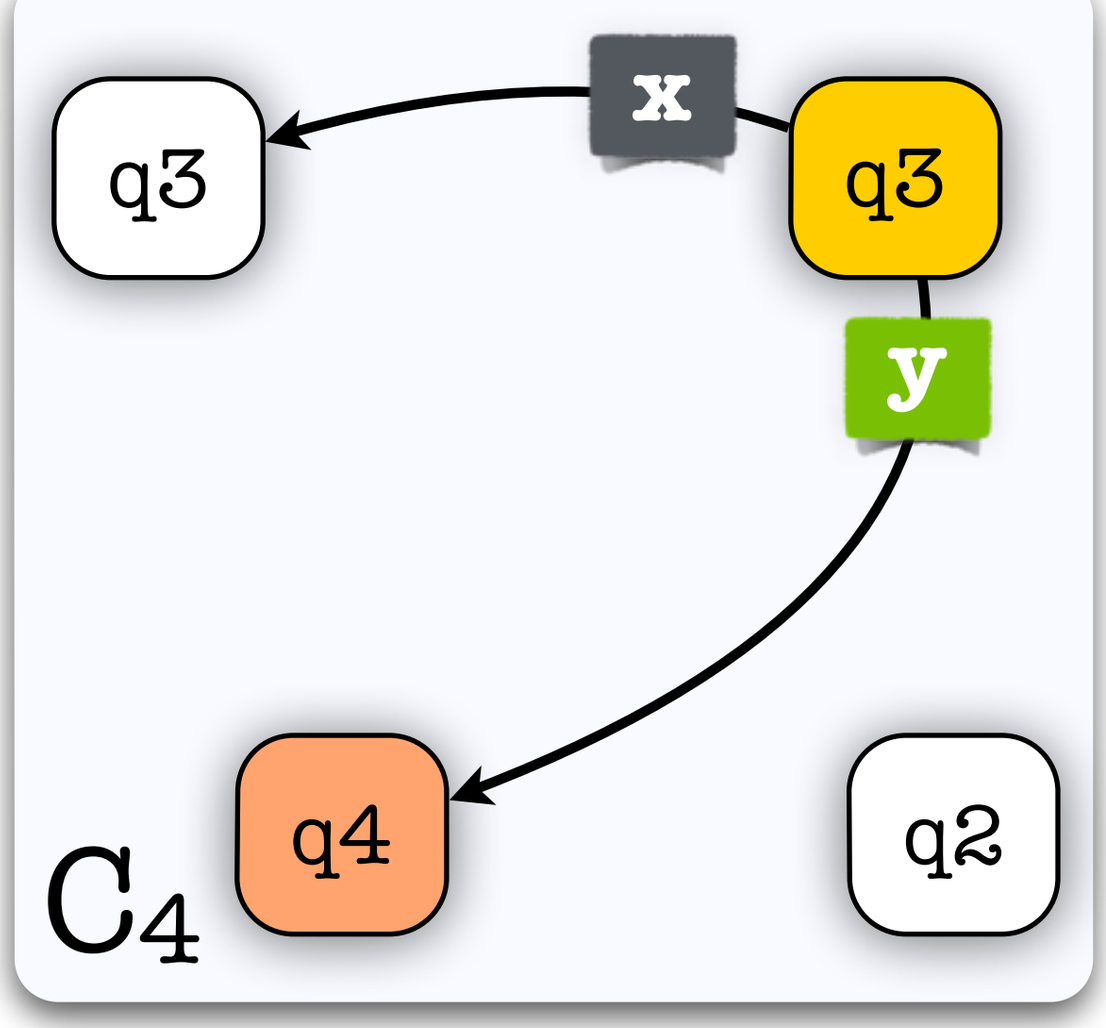
Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Prove **Monotonicity** of Transition Relation

Subgraph relation: Register Mapping & States preserved





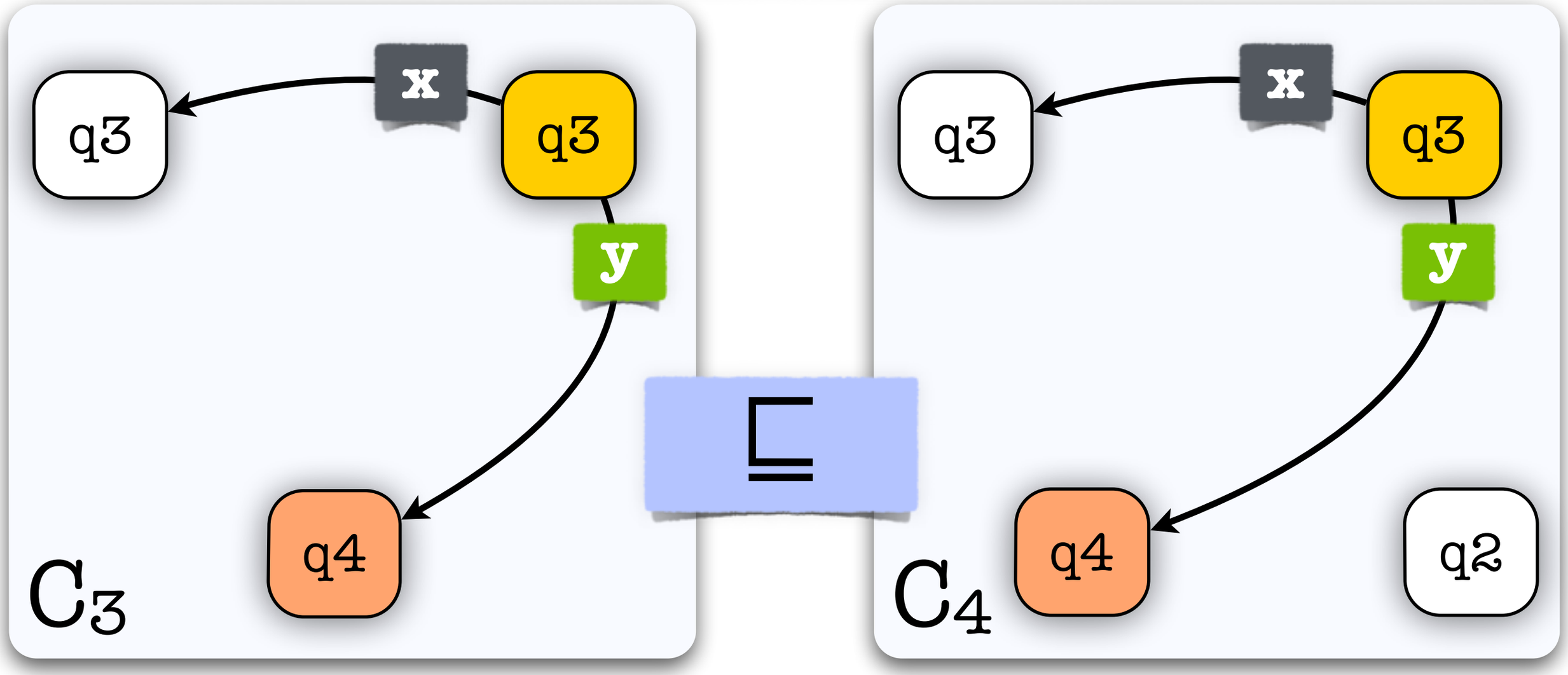
Proofs

Verification of Buffered Dynamic Register Automata

Well-Structured Transition Systems

- Prove **Monotonicity** of Transition Relation

Subgraph relation: Register Mapping & States preserved



Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations
- Prove **Monotonicity** of Transition Relation
- Provide an algorithm to compute the **Pre** of an upward closed set

Proofs

Verification of Buffered Dynamic Register Automata

Lossy + Bound the Buffer + Strongly Bound Simple Path

Well-Structured Transition Systems

- Define a **Well-Quasi Order** on configurations
- Prove **Monotonicity** of Transition Relation
- Provide an algorithm to compute the **Pre** of an upward closed set