Stateful Partial-Order Reduction Methods for Concurrent Systems with Blocking Instructions

Sarah Larroze-Jardiné

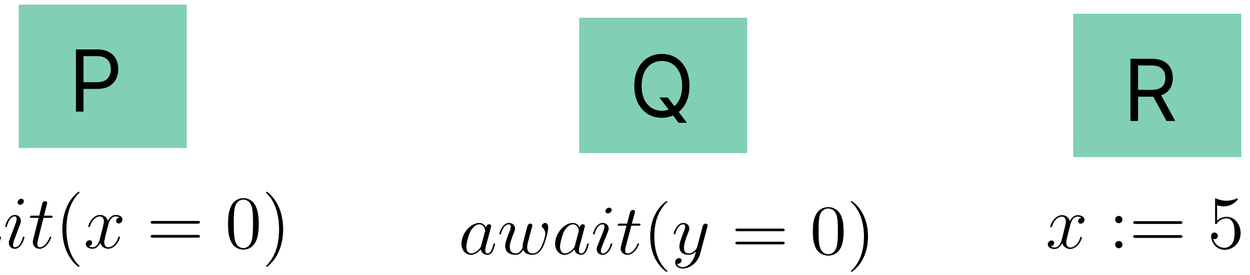
PhD Defense - December 16, 2025, Bordeaux, France

Supervised by Frédéric Herbreteau and Igor Walukiewicz

Verification

Concurrent Programs

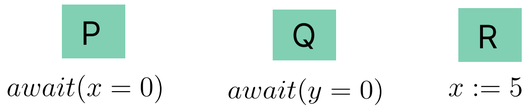
shared int x, y = 0



Verification

Concurrent Programs

shared int x, y = 0



Reachability Problem

Is there a bad state reachable?

Verification

Concurrent Programs

shared int x, y = 0

P

$await(x = 0)$

Q

$await(y = 0)$

R

$x := 5$

Reachability Problem

Is there a bad state reachable?

Model Checking

Given a concurrent program and a property to verify (here reachability), decide whether the property is satisfied or not

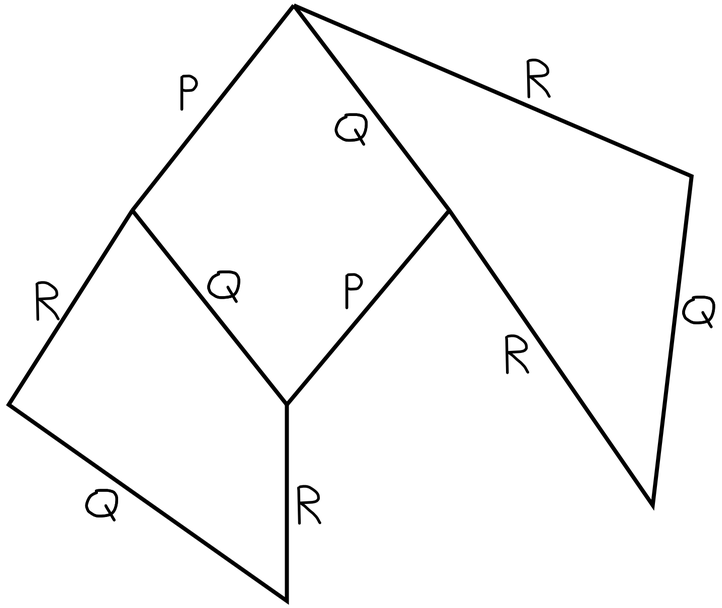
State-space exploration

Program

shared int x, y = 0

P	Q	R
$await(x = 0)$	$await(y = 0)$	$x := 5$

Execution of the program



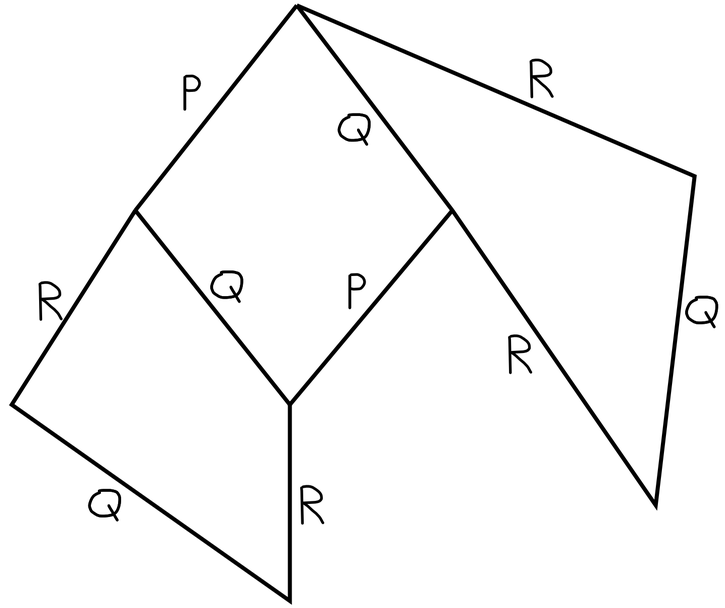
State-space exploration

Program

shared int x, y = 0

P	Q	R
$await(x = 0)$	$await(y = 0)$	$x := 5$

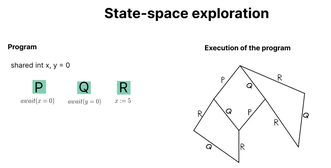
Execution of the program



COMBINATORIAL EXPLOSION!

Model Checking

Given a concurrent program and a property to verify (here reachability), decide whether the property is satisfied or not



COMBINATORIAL EXPLOSION!

Mazurkiewicz’s trace equivalence

independency, equivalence, explore at least one run per equivalence class

Model Checking

Given a concurrent program and a property to verify (here reachability), decide whether the property is satisfied or not

State-space exploration

Program

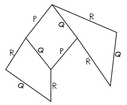
shared int x, y = 0

P — *await*(x = 0)

Q — *await*(y = 0)

R — x := 5

Execution of the program



COMBINATORIAL EXPLOSION!

Mazurkiewicz's trace equivalence

independency, equivalence, explore at least one run per equivalence class

Partial-order reduction

produces a reduced state space that is sound and complete, literature, notion of optimality

Stateless

Stateless = we only enumerate traces
define optimality for stateless

Partial-order reduction

produces a reduced state space that is sound and complete, literature, notion of optimality

Stateful

Stateful = we store the states
define optimality for stateful

Excellent POR algorithm

Here state what an excellent POR algorithm is

theorem concur25 there is no excellent POR algorithm

Stateful

Stateful = we store the states
define optimality for stateful

Excellent POR algorithm

Here state what an excellent POR algorithm is

theorem concur25 there is no excellent POR algorithm

Partial-order reduction

produces a reduced state space that is sound and complete, literature, notion of optimality

Stateless

Stateful

Stateful = we store the states
define optimality for stateful

Excellent POR algorithm

Model Checking

Given a concurrent program and a property to verify (here reachability), decide whether the property is satisfied or not

State-space exploration

Program

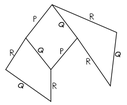
shared int x, y = 0

P
await$(x = 0)$

Q
await$(y = 0)$

R
$x := 5$

Execution of the program



COMBINATORIAL EXPLOSION!

Mazurkiewicz's trace equivalence

independency, equivalence, explore at least one run per equivalence class

Partial-order reduction

produces a reduced state space that is sound and complete, literature, notion of optimality

Verification

Concurrent Programs

shared int x, y = 0

P

$await(x = 0)$

Q

$await(y = 0)$

R

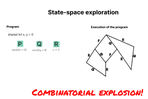
$x := 5$

Reachability Problem

Is there a bad state reachable?

Model Checking

Given a concurrent program and a property to verify (here reachability), decide whether the property is satisfied or not



Stateful Partial-Order Reduction Methods for Concurrent Systems with Blocking Instructions

Sarah Larroze-Jardiné

PhD Defense - December 16, 2025, Bordeaux, France

Supervised by Frédéric Herbreteau and Igor Walukiewicz

LaBRI

université de BORDEAUX

Contributions of this thesis

- client/server systems as a unified framework
- POR graph with covering, i.e. all ingredients of our algorithms
- lexicographic stateful por
- por with race reversals
- Implementation and experiments

Client/server systems

show an example and give the intuition that a client represents a shared ressource

Contributions of this thesis

- client/server systems as a unified framework
- POR graph with covering, i.e. all ingredients of our algorithms
- lexicographic stateful por
- por with race reversals
- Implementation and experiments

POR graphs with covering

Source sets

notion of first
exploration with source sets sound and complete

POR graphs with covering

Source sets

notion of first
exploration with source sets sound and complete

Sleep sets

Introduce sleep sets
Pb of sleep blocked execution

POR graphs with covering

Sleep sets

Introduce sleep sets
Pb of sleep blocked execution

Source sets

notion of first
exploration with source sets sound and complete

Node subsumption

example with node duplication
give intuition why it is a simulation relation

POR graphs with covering

Sleep sets

Introduce sleep sets
Pb of sleep blocked execution

Source sets

notion of first
exploration with source sets sound and complete

Node subsumption

example with node duplication
give intuition why it is a simulation relation

Contributions of this thesis

- client/server systems as a unified framework
- POR graph with covering, i.e. all ingredients of our algorithms
- lexicographic stateful por
- por with race reversals
- Implementation and experiments

POR with race reversals

Races

POR with race reversals

Explore not below

POR with race reversals

Races

Explore not below

Explore above

POR with race reversals

Races

Explore not below

Explore above

Not below vs above

POR with race reversals

Contributions of this thesis

- client/server systems as a unified framework
- POR graph with covering, i.e. all ingredients of our algorithms
- lexicographic stateful por
- por with race reversals
- Implementation and experiments

Experiments

Contributions of this thesis

- client/server systems as a unified framework
- POR graph with covering, i.e. all ingredients of our algorithms
- lexicographic stateful por
- por with race reversals
- Implementation and experiments

Stateful Partial-Order Reduction Methods for Concurrent Systems with Blocking Instructions

Sarah Larroze-Jardiné

PhD Defense - December 16, 2025, Bordeaux, France

Supervised by Frédéric Herbreteau and Igor Walukiewicz

LaBRI

université de BORDEAUX

Conclusion and future work

Stateful Partial-Order Reduction Methods for Concurrent Systems with Blocking Instructions

Sarah Larroze-Jardiné

PhD Defense - December 16, 2025, Bordeaux, France

Supervised by Frédéric Herbreteau and Igor Walukiewicz

LaBRI

université de BORDEAUX