Categorical Automata Learning Framework

calf-project.org

Gerco van Heerdt **Matteo Sammartino** Alexandra Silva

LiVe @ ETAPS 2017

UCL ENGINEERING
Change the world

UCL

Our team

Alexandra Silva
UCL

Matteo Sammartino
UCL

Gerco van Heerdt
UCL

Joshua Moerman
Radboud University

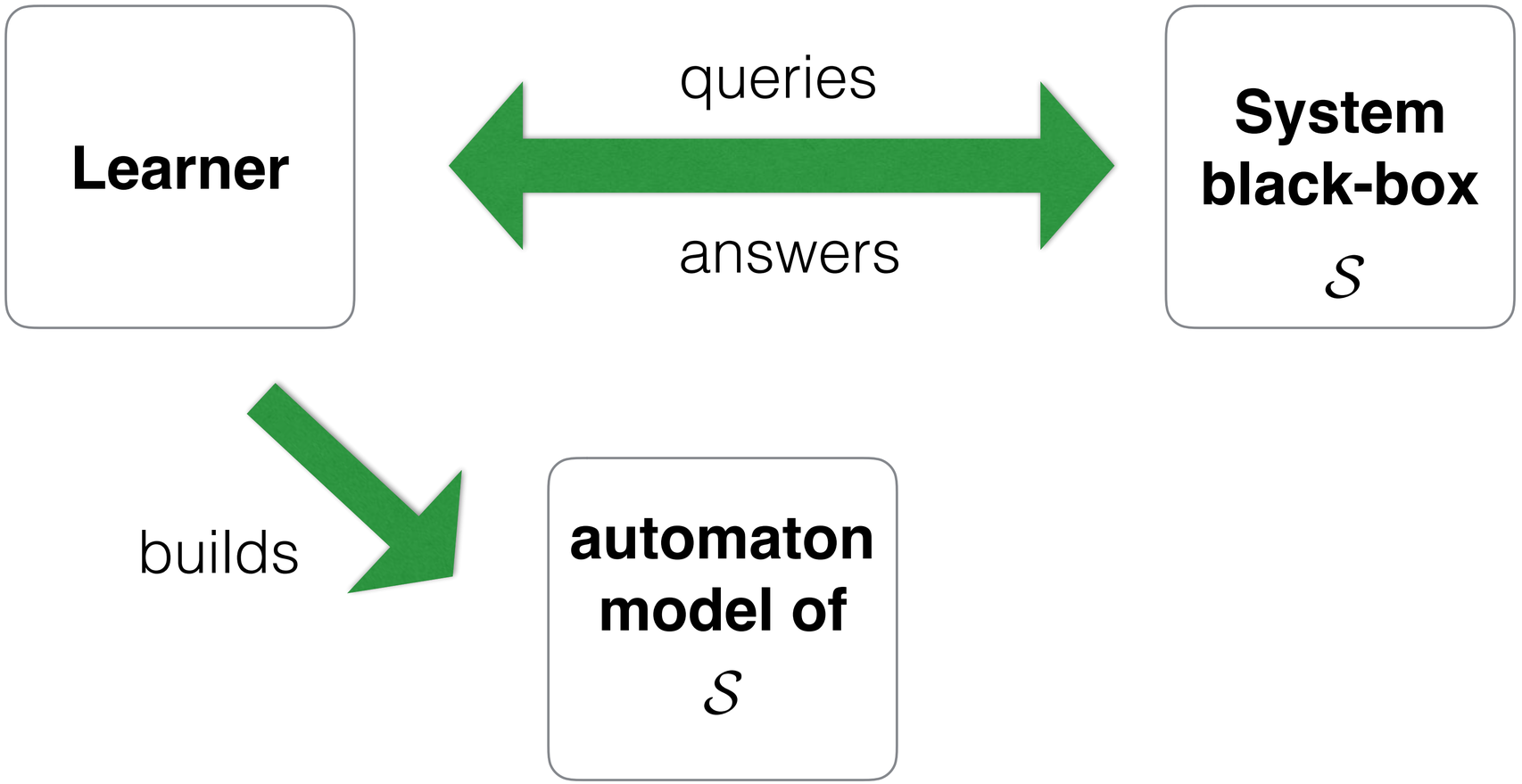
Maverick Chardet
ENS Lyon

Collaborators:

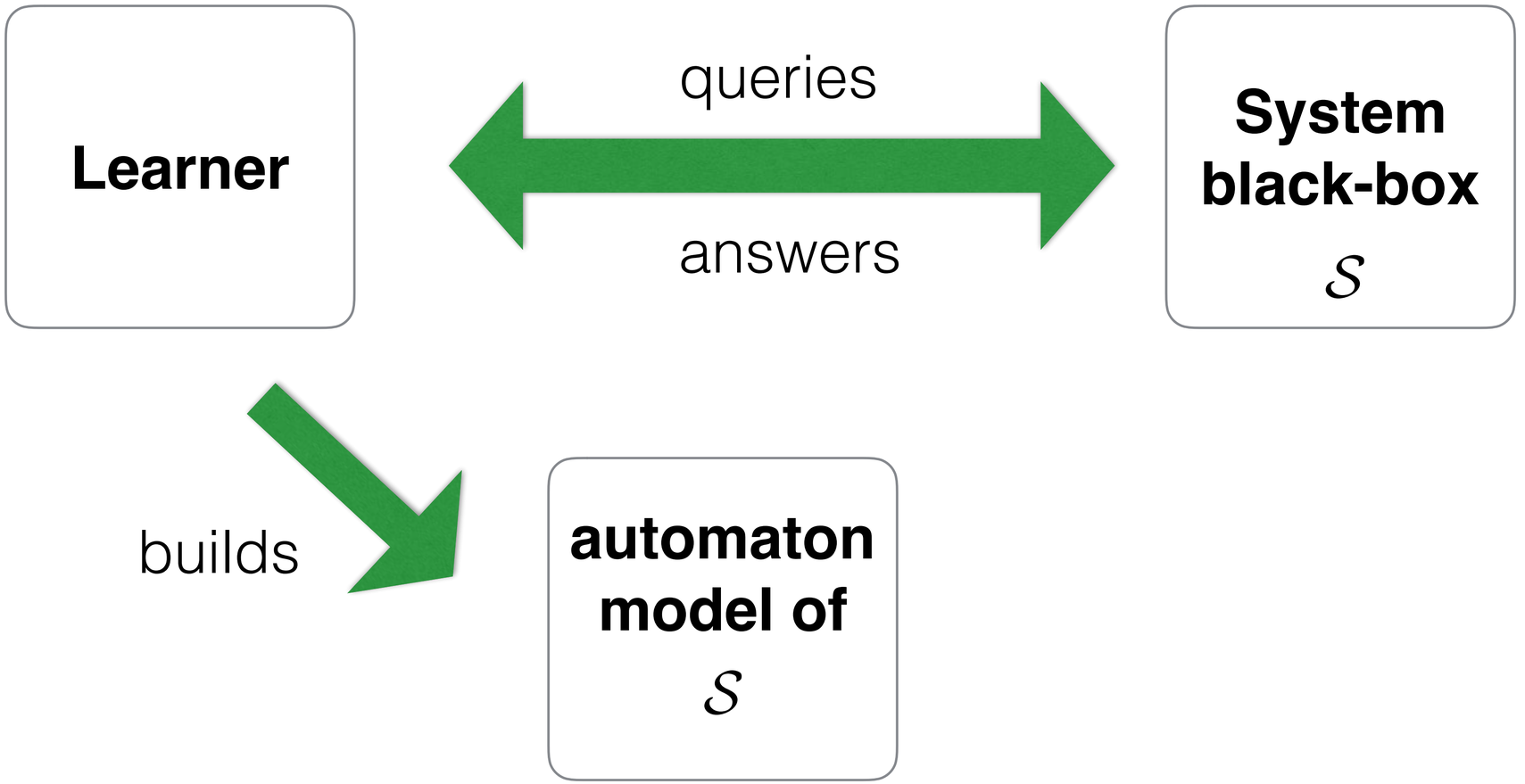
Bartek Klin
Warsaw University

Michal Szynwelski
Warsaw University

Automata learning



Automata learning



No formal specification available? **Learn it!**

L* algorithm (D.Angluin ’87)

Finite alphabet of system’s actions A

set of system behaviors is a **regular language** $\mathcal{L} \subseteq A^\star$

L* algorithm (D.Angluin ’87)

Finite alphabet of system’s actions A

set of system behaviors is a **regular language** $\mathcal{L} \subseteq A^\star$

Learner

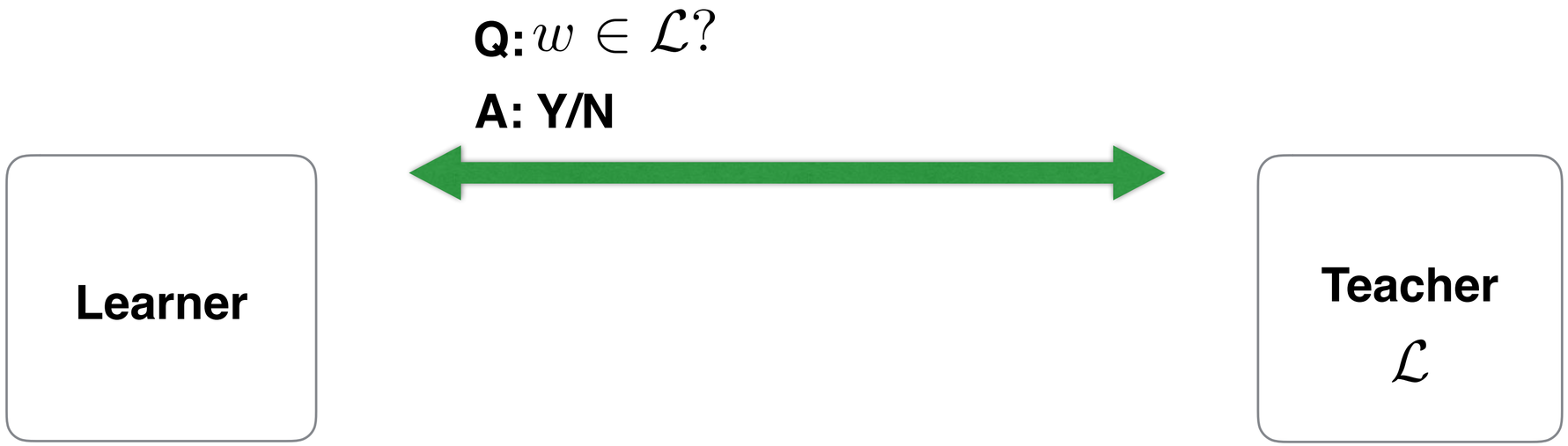
Teacher

$\mathcal{L}$

L* algorithm (D.Angluin ’87)

Finite alphabet of system’s actions A

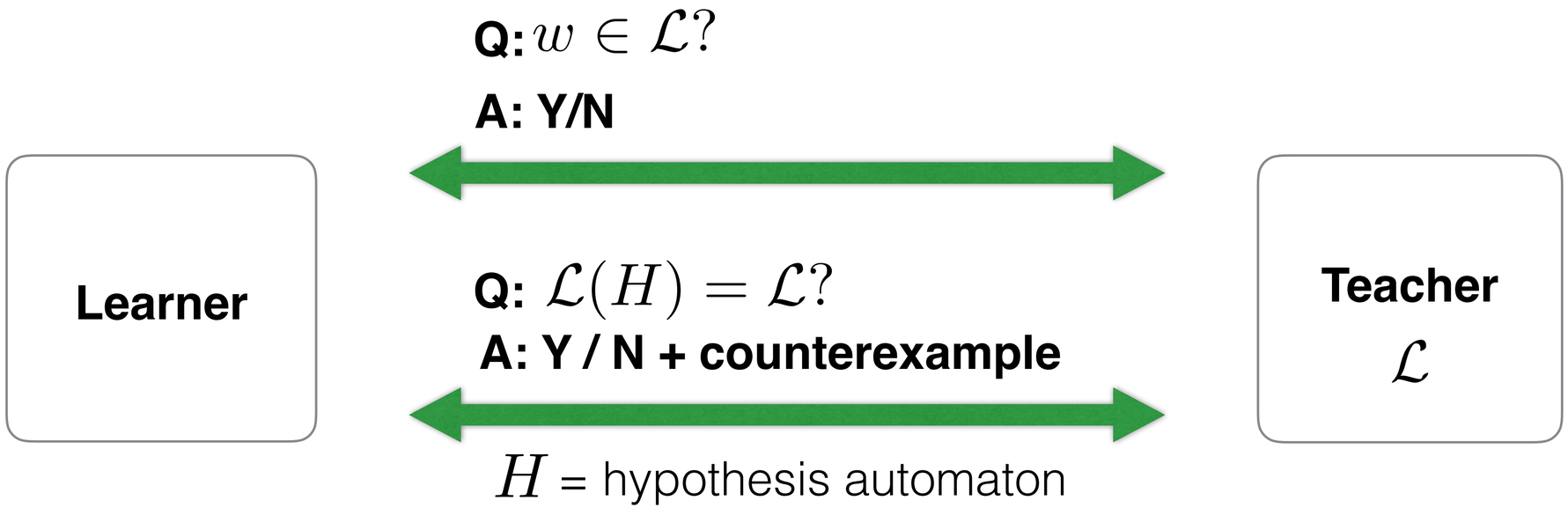
set of system behaviors is a **regular language** $\mathcal{L} \subseteq A^\star$



L* algorithm (D.Angluin ’87)

Finite alphabet of system’s actions A

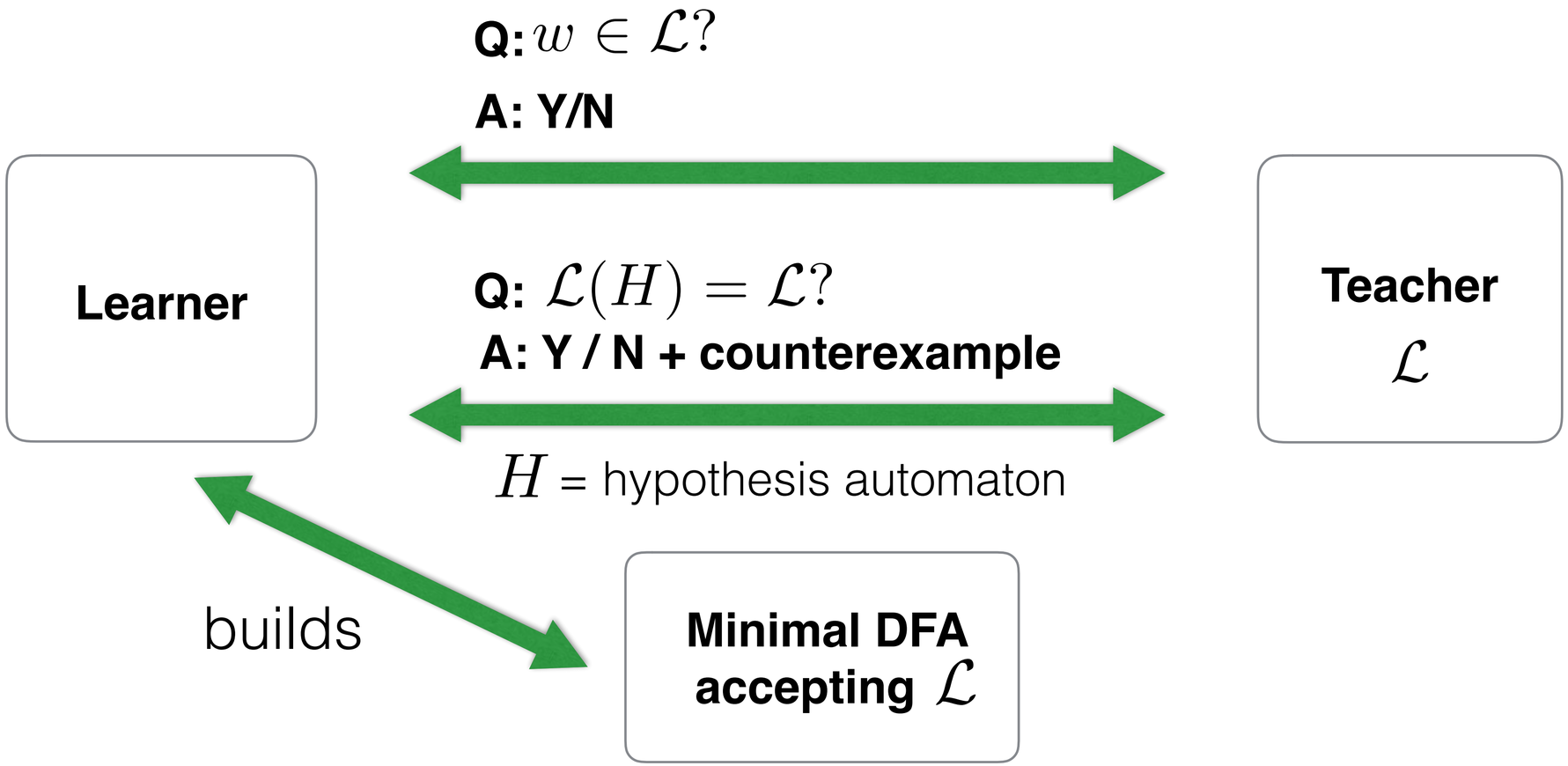
set of system behaviors is a **regular language** $\mathcal{L} \subseteq A^\star$



L* algorithm (D.Angluin ’87)

Finite alphabet of system’s actions A

set of system behaviors is a **regular language** $\mathcal{L} \subseteq A^\star$



A zoo of automata

Probabilistic

Weighted

Alternating

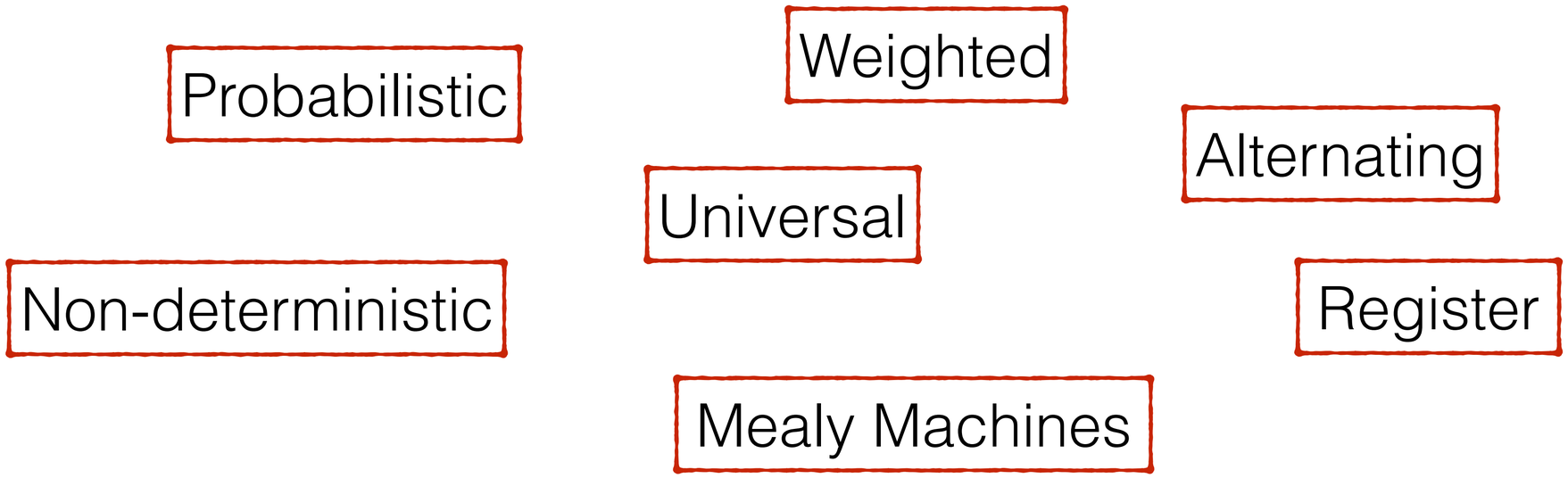
Universal

Non-deterministic

Register

Mealy Machines

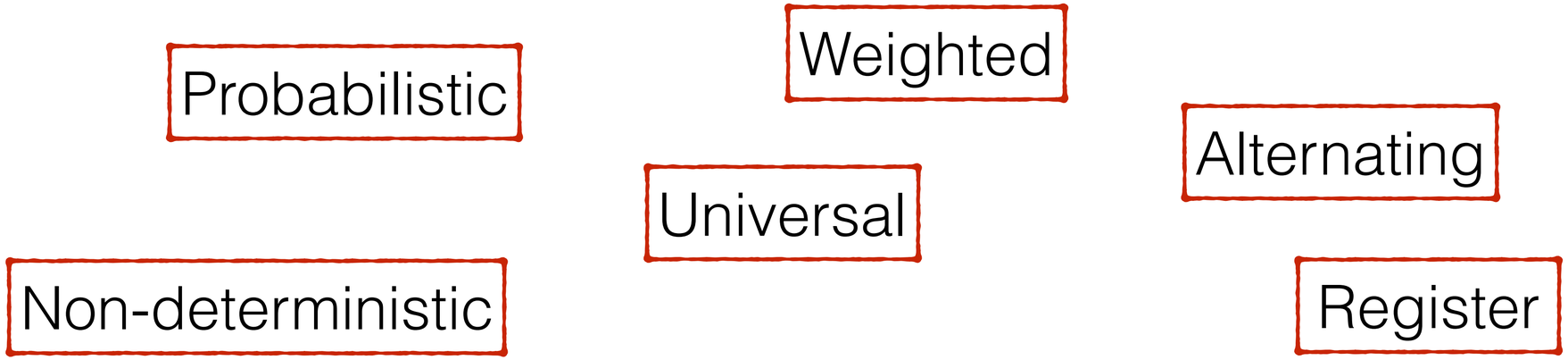
A zoo of automata



Algorithms **Correctness proofs**

involved and hard to check

A zoo of automata



Category theory comes to the rescue!

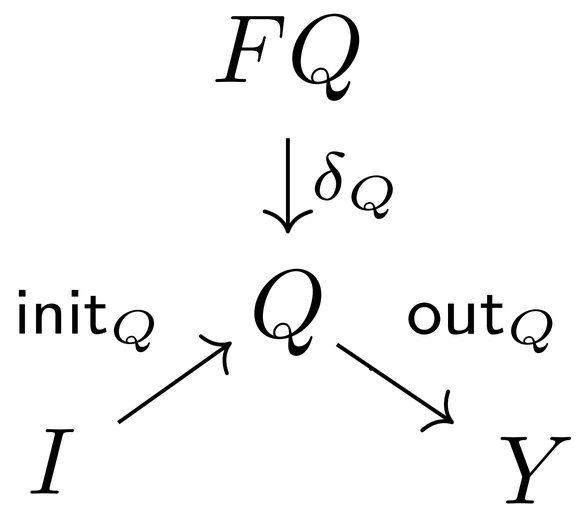
Algorithms **Correctness proofs**

involved and hard to check

Abstract automata

Category $\mathbf{C}$ = universe of state-spaces

Endofunctor $F\colon \mathbf{C} \to \mathbf{C}$ = automaton type



Abstract automata

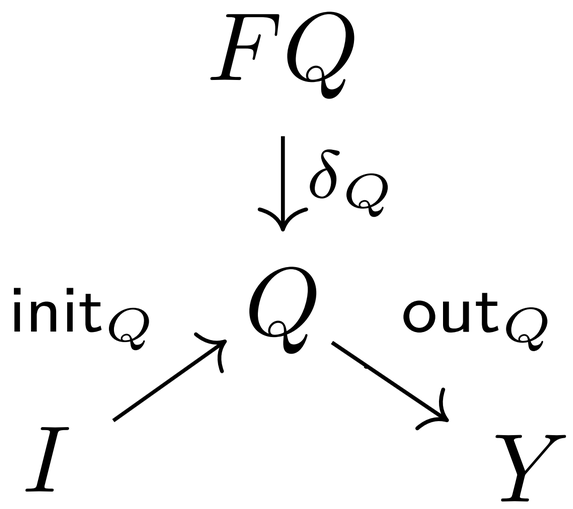
Category $\mathbf{C}$ = universe of state-spaces

Endofunctor $F : \mathbf{C} \to \mathbf{C}$ = automaton type

DFAs

$\mathbf{C} = \mathbf{Set}$

$F = (-) \times A$



Abstract automata

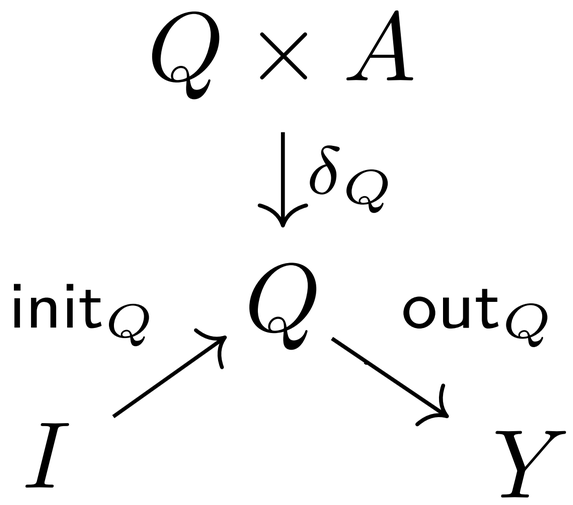
Category $\mathbf{C}$ = universe of state-spaces

Endofunctor $F : \mathbf{C} \to \mathbf{C}$ = automaton type

DFAs

$\mathbf{C} = \mathbf{Set}$

$F = (-) \times A$



Abstract automata

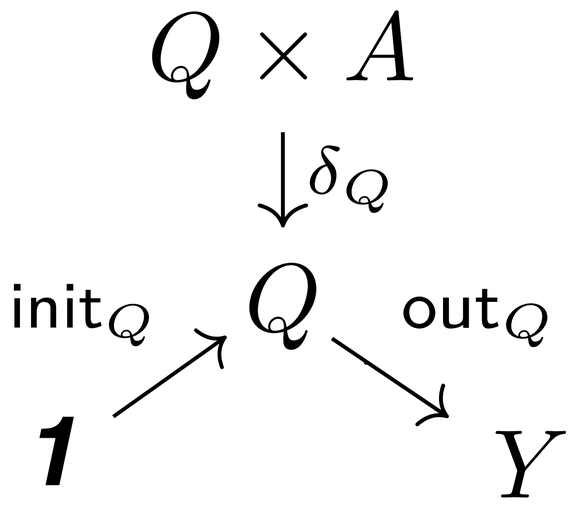
Category $\mathbf{C}$ = universe of state-spaces

Endofunctor $F : \mathbf{C} \to \mathbf{C}$ = automaton type

DFAs

$\mathbf{C} = \mathbf{Set}$

$F = (-) \times A$



Abstract automata

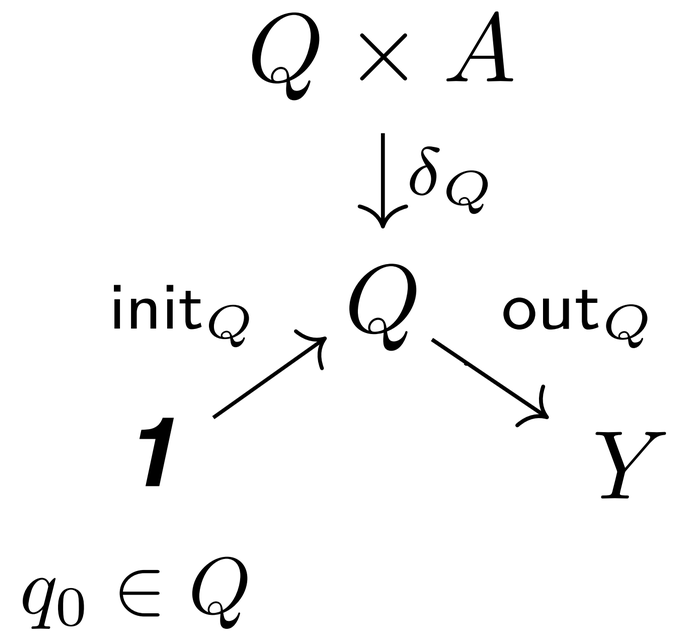
Category $\mathbf{C}$ = universe of state-spaces

Endofunctor $F : \mathbf{C} \to \mathbf{C}$ = automaton type

DFAs

$\mathbf{C} = \mathbf{Set}$

$F = (-) \times A$



$q_0 \in Q$

Abstract automata

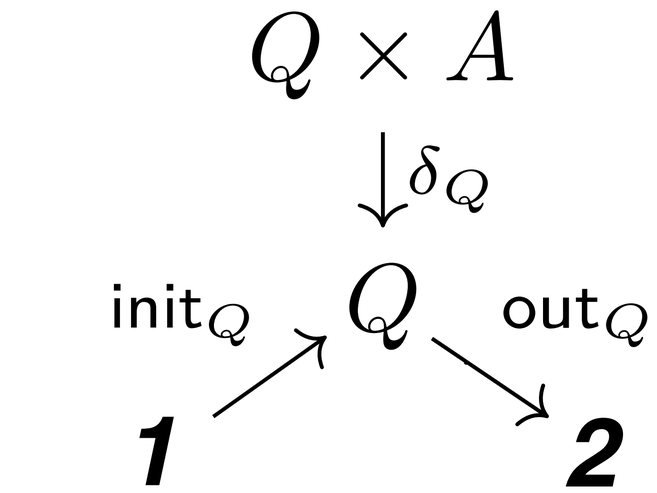
Category $\mathbf{C}$ = universe of state-spaces

Endofunctor $F : \mathbf{C} \to \mathbf{C}$ = automaton type

DFAs

$\mathbf{C} = \mathbf{Set}$

$F = (-) \times A$



$q_0 \in Q$

Abstract automata

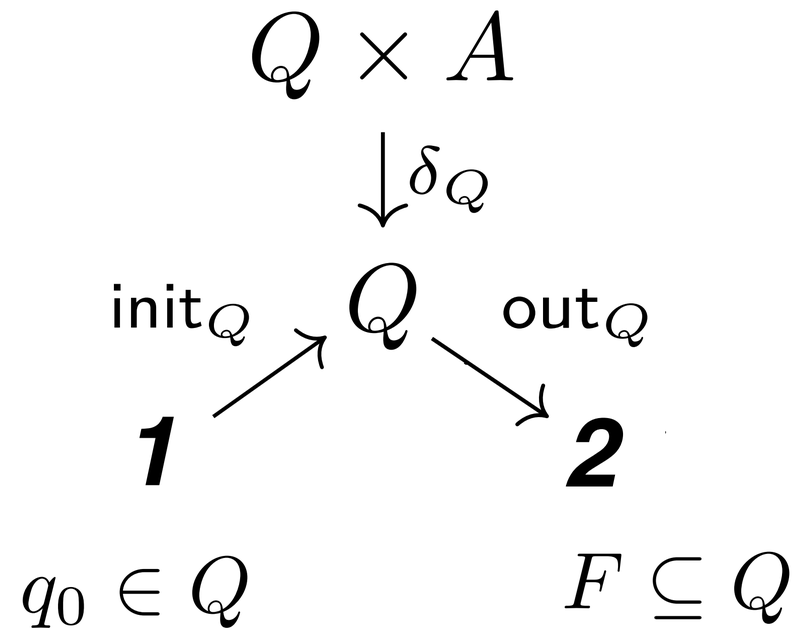
Category $\mathbf{C}$ = universe of state-spaces

Endofunctor $F : \mathbf{C} \to \mathbf{C}$ = automaton type

DFAs

$\mathbf{C} = \mathbf{Set}$

$F = (-) \times A$



Abstract learning

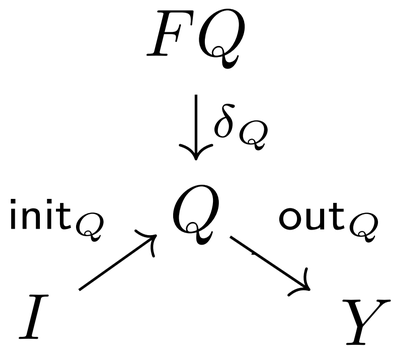
Abstract observation data structure

Abstract learning

Abstract observation data structure

approximates

Target minimal automaton

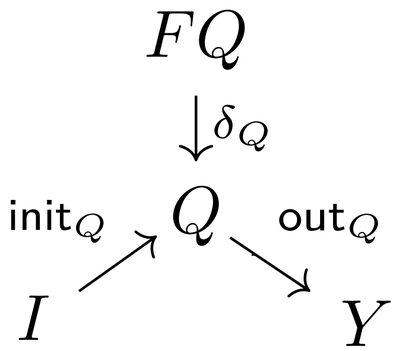


Abstract learning

Abstract observation data structure

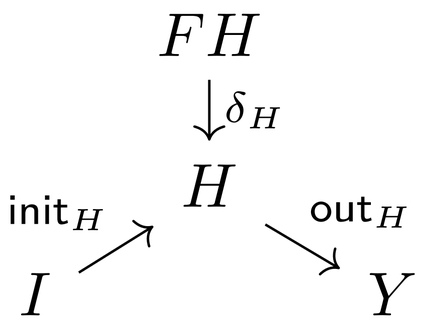
approximates

Target minimal automaton



abstract closedness and consistency

Hypothesis automaton

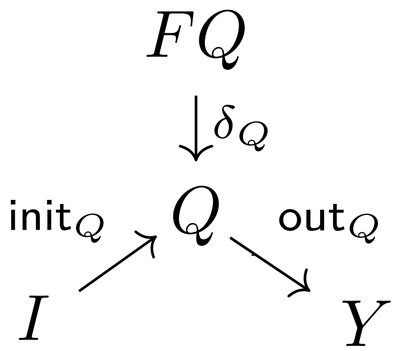


Abstract learning

Abstract observation data structure

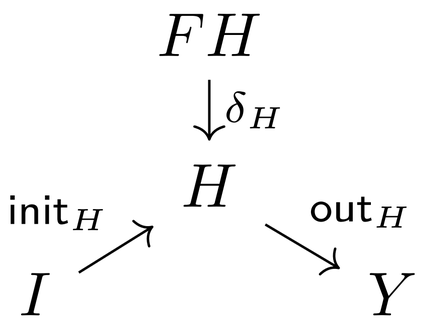
approximates

Target minimal automaton



abstract closedness and consistency

Hypothesis automaton



General correctness theorem

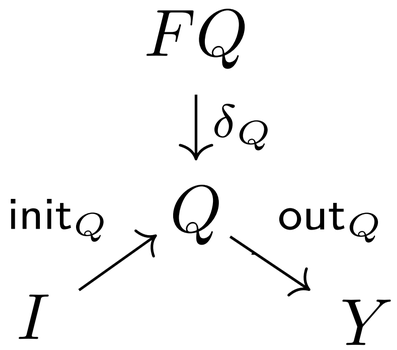
Guidelines for implementation

Abstract learning

Abstract observation data structure

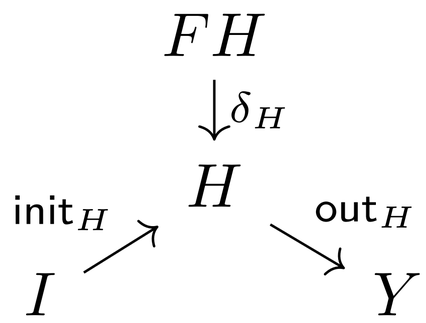
approximates

Target minimal automaton



abstract closedness and consistency

Hypothesis automaton



General correctness theorem

Guidelines for implementation

CALF: Categorical Automata Learning Framework (arXiv:1704.05676)

Gerco van Heerdt, Matteo Sammartino, Alexandra Silva

Other automata & optimizations

Other automata & optimizations

Change base category

Set	**DFAs**
Nom	**Nominal automata**
Vect	**Weighted automata**

Other automata & optimizations

Change base category

Set	**DFAs**
Nom	**Nominal automata**
Vect	**Weighted automata**

Side-effects (via monads)

Powerset	**NFAs**
Powerset with intersection	**Universal automata**
Double powerset	**Alternating automata**
Maybe monad	**Partial automata**

Other automata & optimizations

Change base category

Set	**DFAs**
Nom	**Nominal automata**
Vect	**Weighted automata**

Change main data structure

Observation tables

Discrimination trees

Side-effects (via monads)

Powerset	**NFAs**
Powerset with intersection	**Universal automata**
Double powerset	**Alternating automata**
Maybe monad	**Partial automata**

Other automata & optimizations

Change base category

Set	**DFAs**
Nom	**Nominal automata**
Vect	**Weighted automata**

Change main data structure

Learning Nominal Automata (POPL ’17)

Joshua Moerman, Matteo Sammartino, Alexandra Silva, Bartek Klin, Michal Szynwelski

Discrimination trees

Side-effects (via monads)

Powerset	**NFAs**
Powerset with intersection	**Universal automata**
Double powerset	**Alternating automata**
Maybe monad	**Partial automata**

Other automata & optimizations

Change base category

Set	**DFAs**
Nom	**Nominal automata**
Vect	**Weighted automata**

Change main data structure

Learning Nominal Automata (POPL ’17)

Joshua Moerman, Matteo Sammartino, Alexandra Silva, Bartek Klin, Michal Szynwelski

Discrimination trees

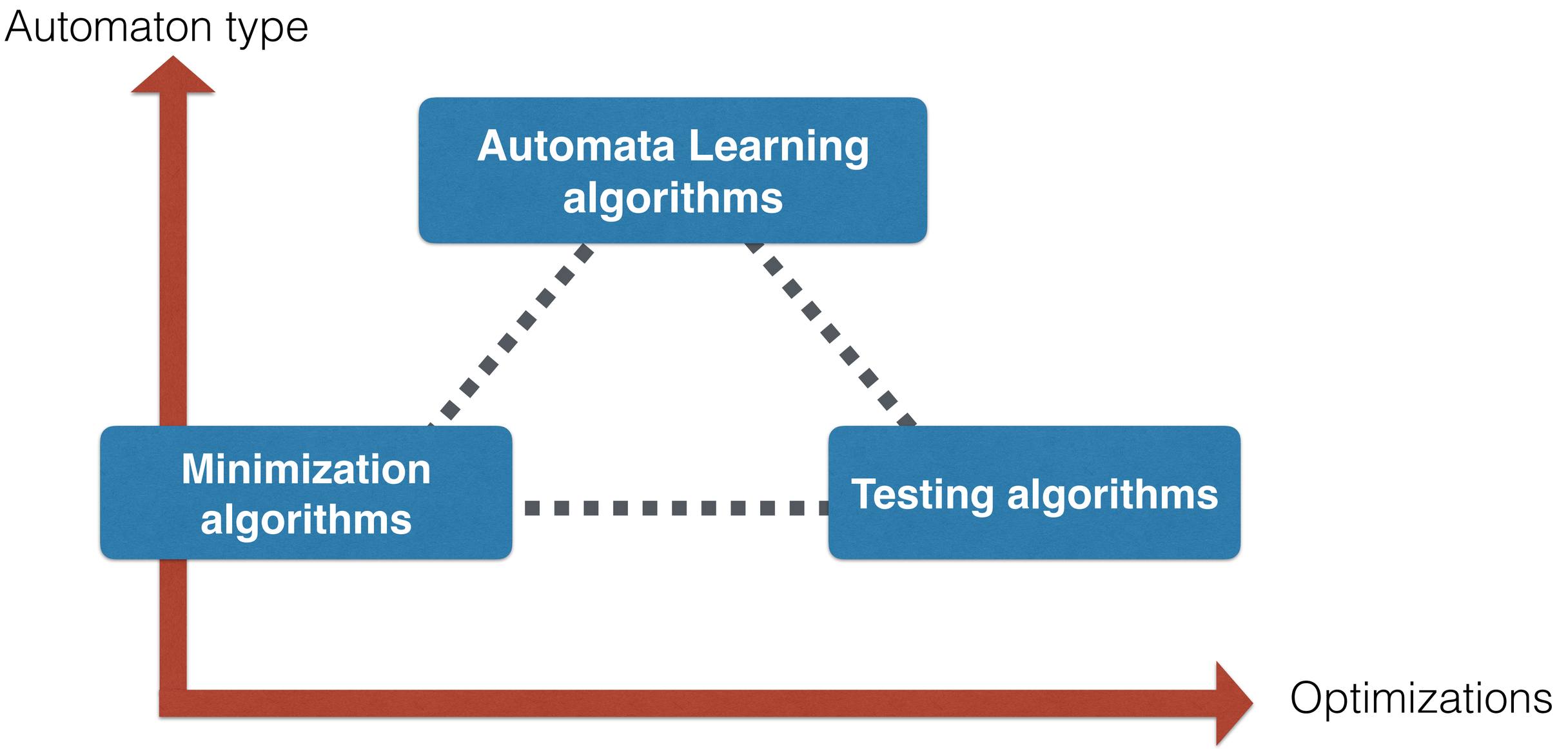
Learning Automata with Side-effects (arXiv:1704.08055)

Gerco van Heerdt, Matteo Sammartino, Alexandra Silva

Side-effects (via monads)

Powerset	**NFAs**
Powerset with intersection	**Universal automata**
Double powerset	**Alternating automata**
Maybe monad	**Partial automata**

Connections with other algorithms



Ongoing and future work

- **Library & tool** to learn control + data-flow models (as **nominal automata**)
- Applications:
  - Specification mining
  - Network verification, with amazon
  - Verification of cryptographic protocols
  - Ransomware detection