Continuous One-Counter Automata

Philip Offtermatt

Joint work with:
Michael Blondin, Tim Leys,
Filip Mazowiecki, Guillermo Pérez

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Continuous One-Counter Automata

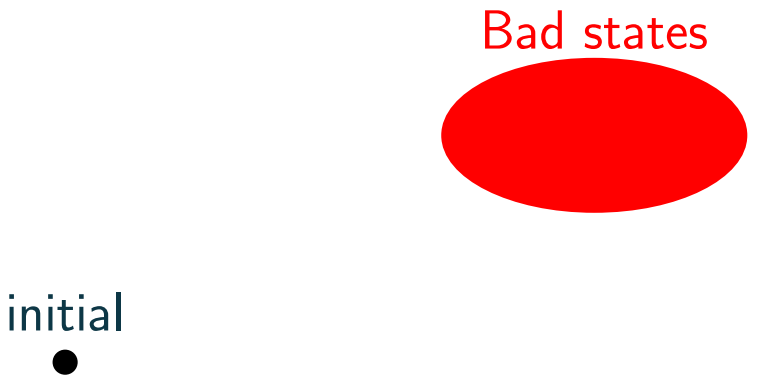
Philip Offtermatt

Joint work with:
Michael Blondin, Tim Leys,
Filip Mazowiecki, Guillermo Pérez

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

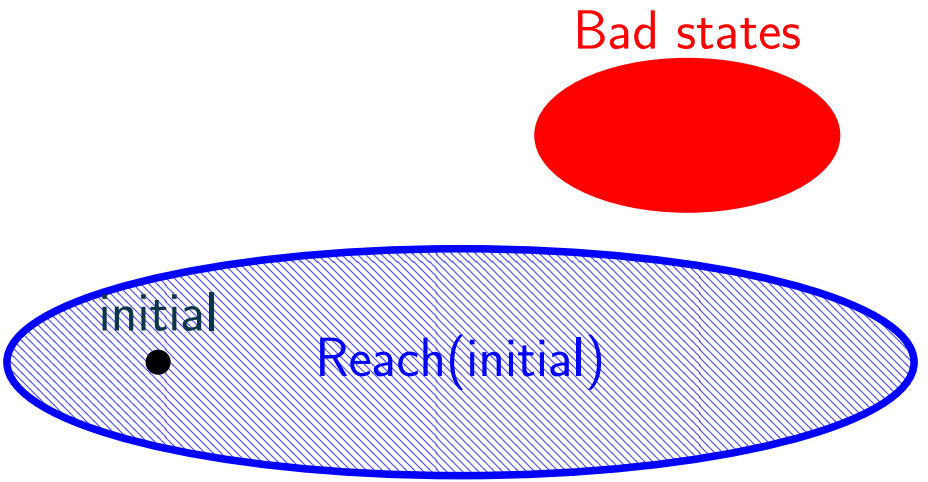
Why we care about overapproximations

Goal: Verify no **bad state** is **reachable**!



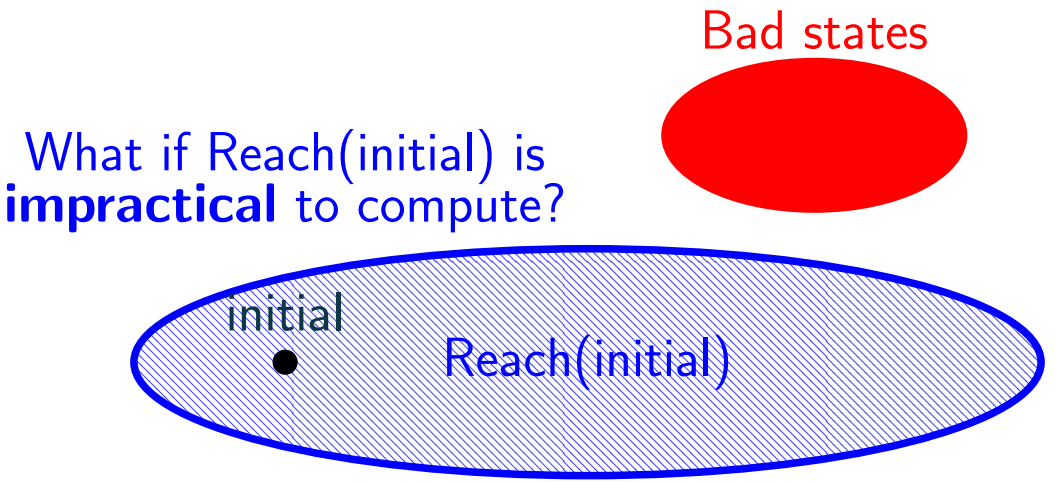
Why we care about overapproximations

Goal: Verify no **bad state** is **reachable**!



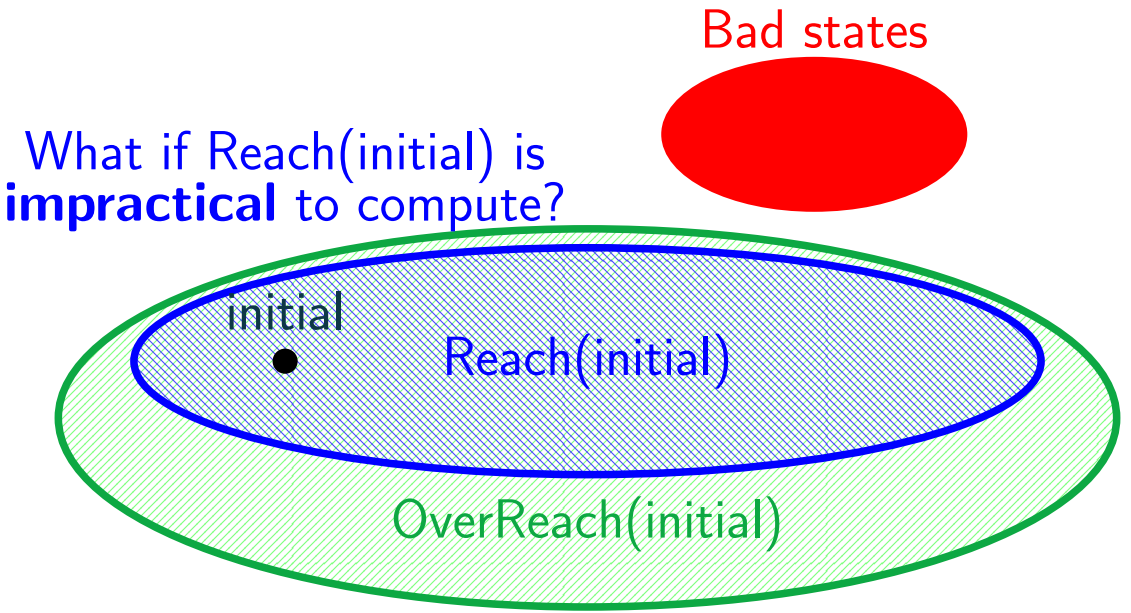
Why we care about overapproximations

Goal: Verify no **bad state** is **reachable**!



Why we care about overapproximations

Goal: Verify no **bad state** is **reachable**!



Compute an **Overapproximation** instead!

Overapproximations help proving safety

Overapproximations help proving safety

Goal: Find efficient overapproximations for models with hard reachability problems

Overapproximations help proving safety

Goal: Find efficient overapproximations
for models with hard reachability problems

Overapproximations help proving safety

Goal: Find efficient overapproximations
for ~~models with hard reachability problems~~
models representing interesting systems

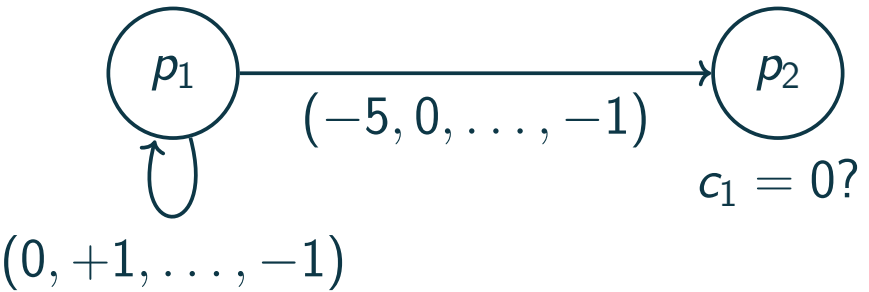
Overapproximations help proving safety

Goal: Find efficient overapproximations
for ~~models with hard reachability problems~~
~~models representing interesting systems~~

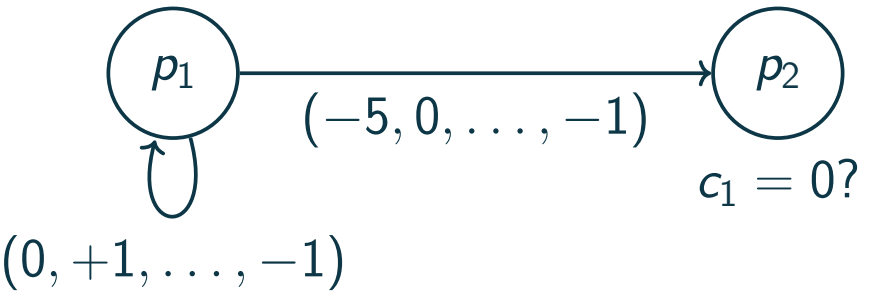
Overapproximations help proving safety

Goal: Find efficient overapproximations
for ~~models with hard reachability problems~~
~~models representing interesting systems~~
One-Counter Automata

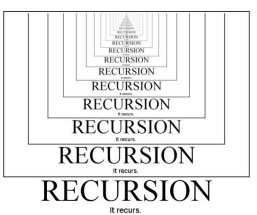
Counter Automata



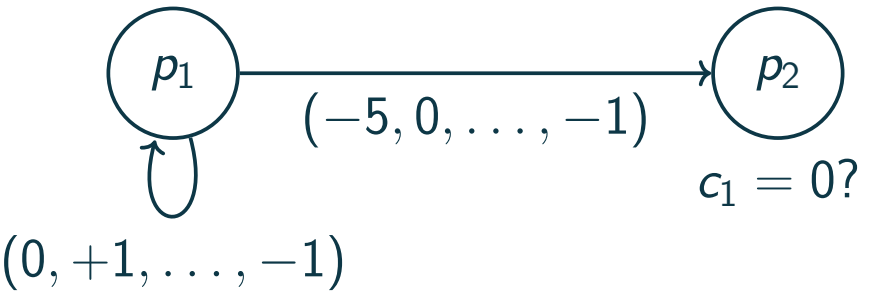
Counter Automata



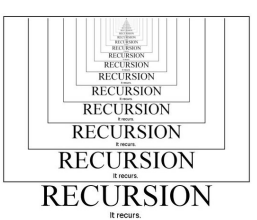
Represent complex systems



Counter Automata



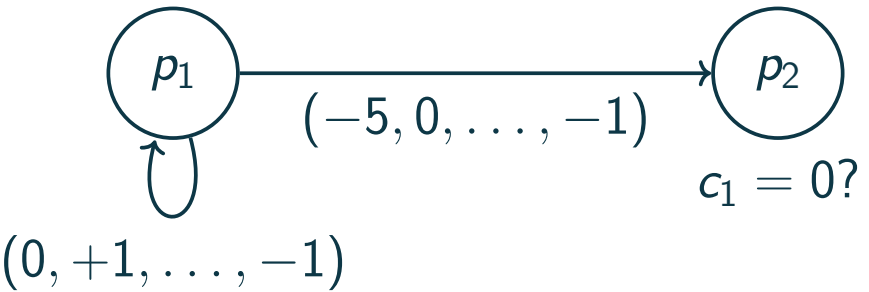
Represent complex systems



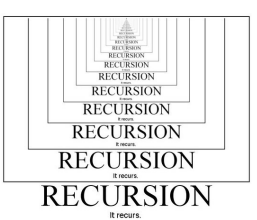
But **Reachability is undecidable!**

...even with only two counters [Minsky,'61]

Counter Automata



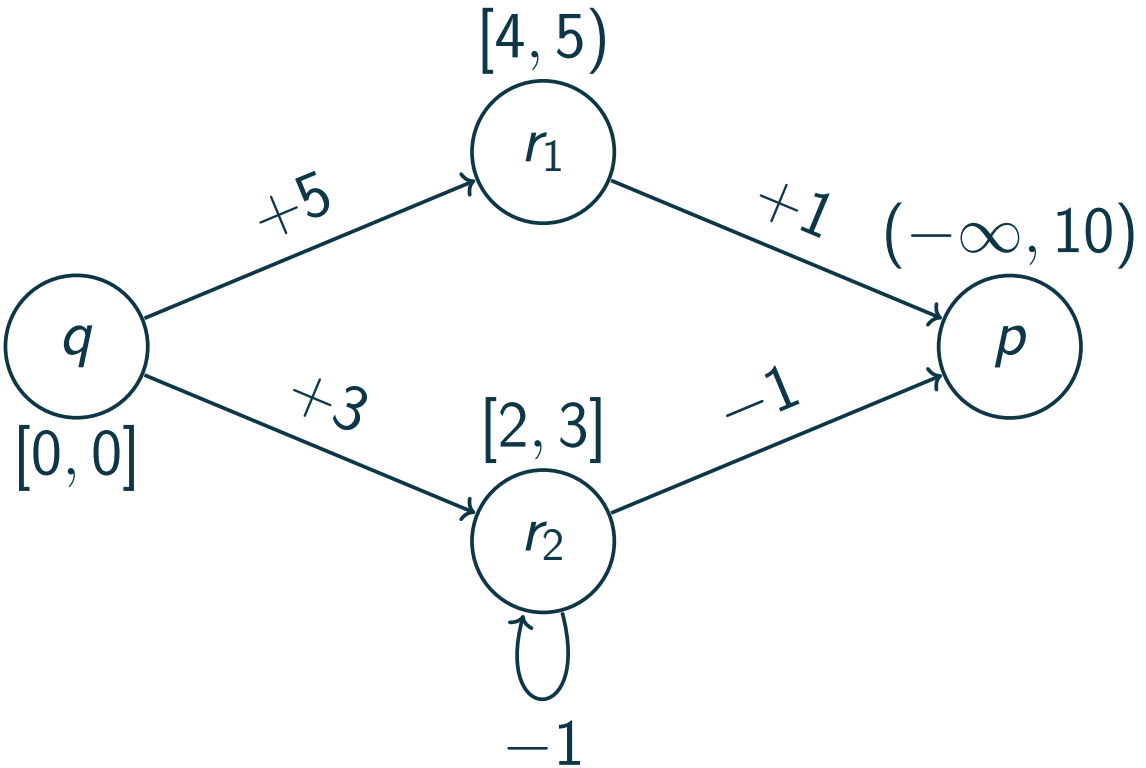
Represent complex systems



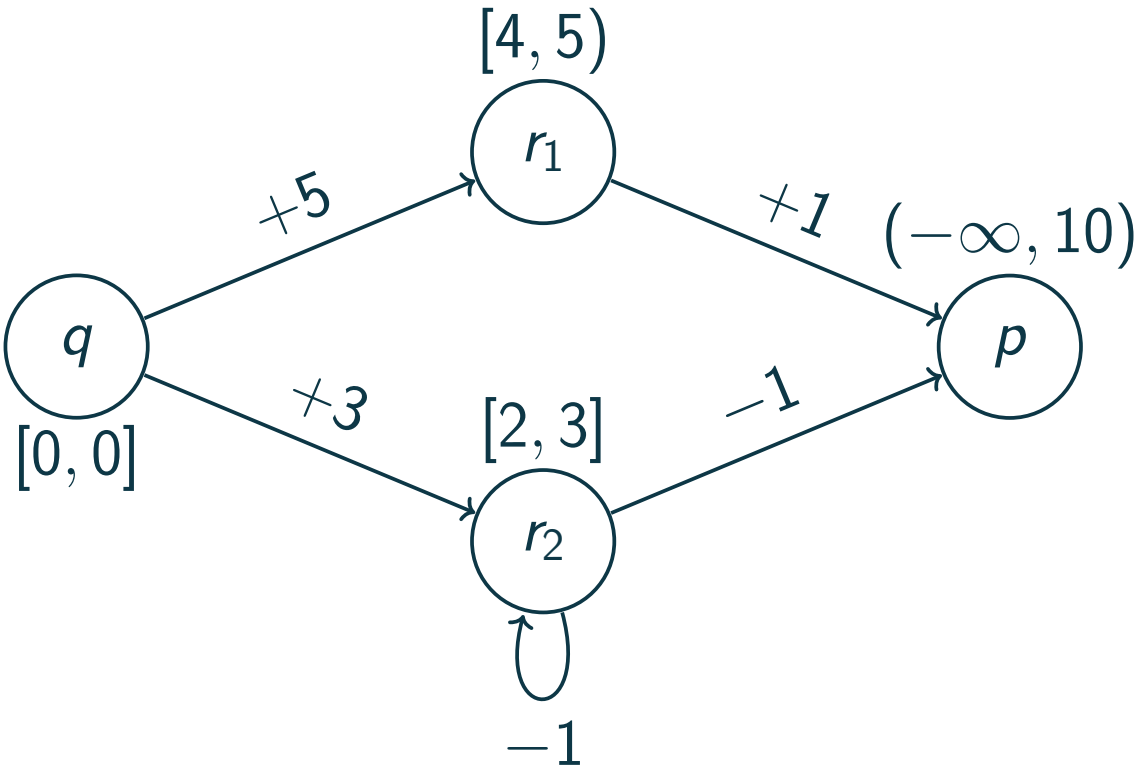
But **Reachability is undecidable!**
...even with only two counters [Minsky,'61]

$\Rightarrow$ Restrict to **One**-Counter Automata!

One-Counter Automata (OCA)

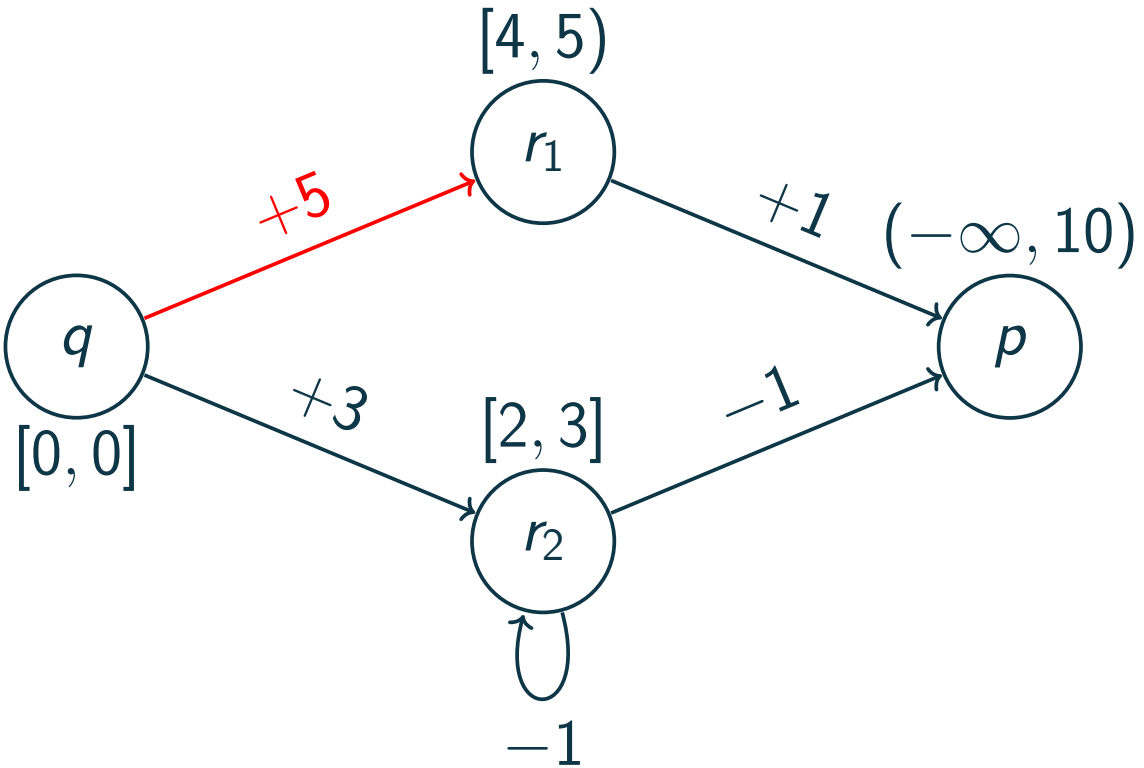


One-Counter Automata (OCA)



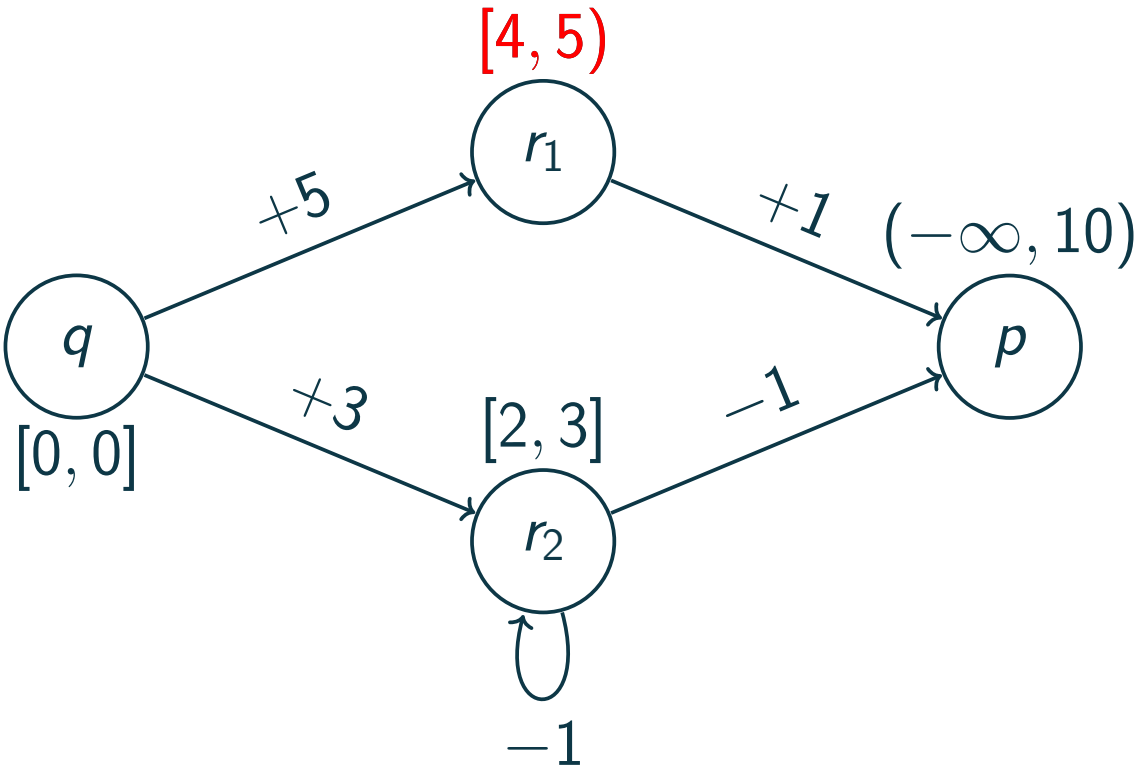
Run: $q(0)$

One-Counter Automata (OCA)



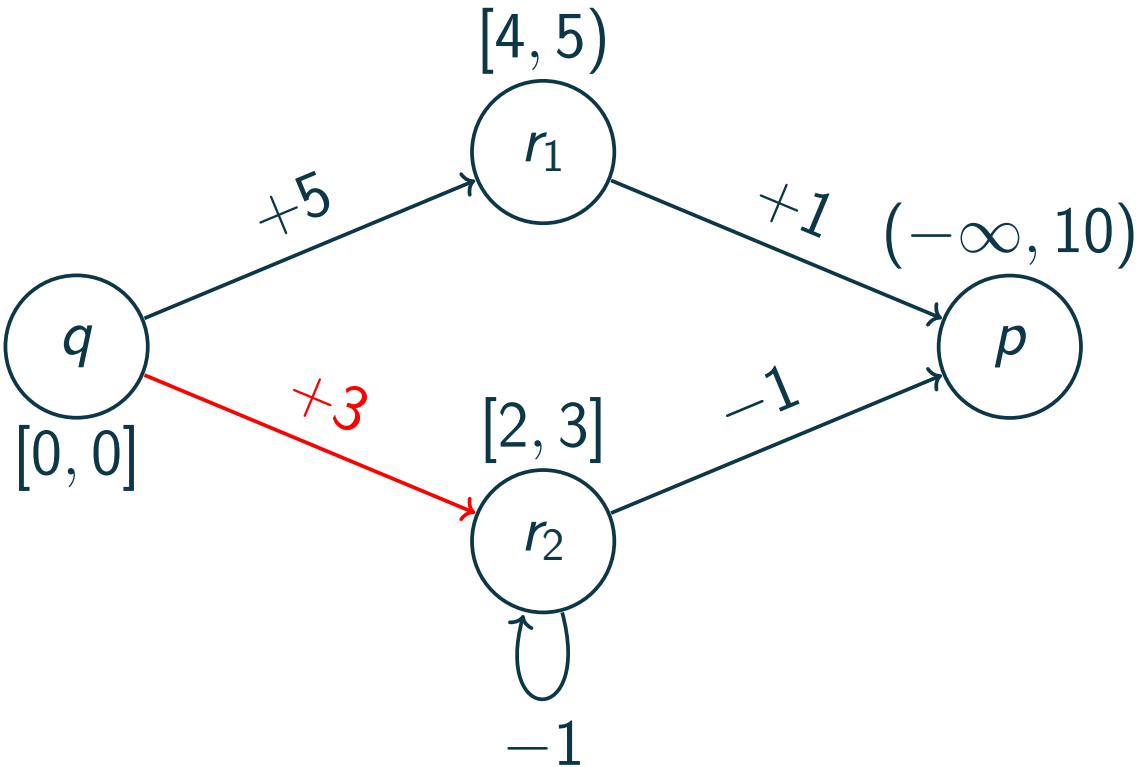
Run: $q(0) \rightarrow r_1(5)$

One-Counter Automata (OCA)



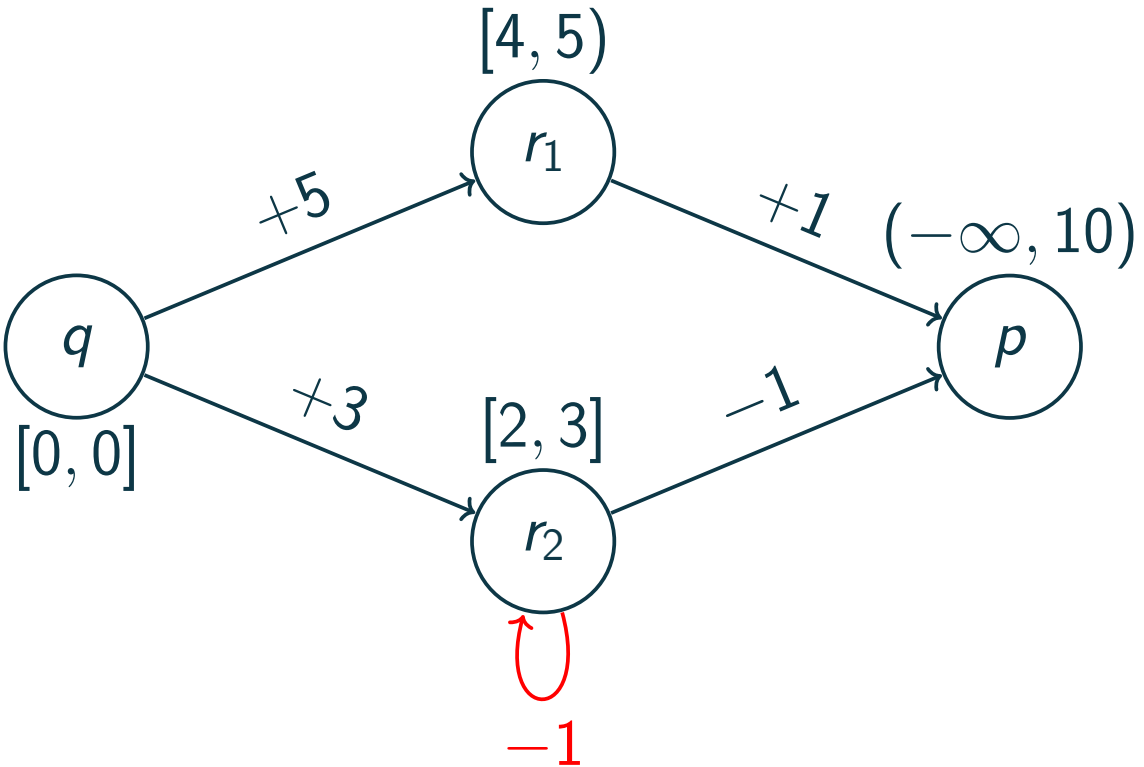
Run: $q(0) \rightarrow \cancel{r_1(5)}$

One-Counter Automata (OCA)



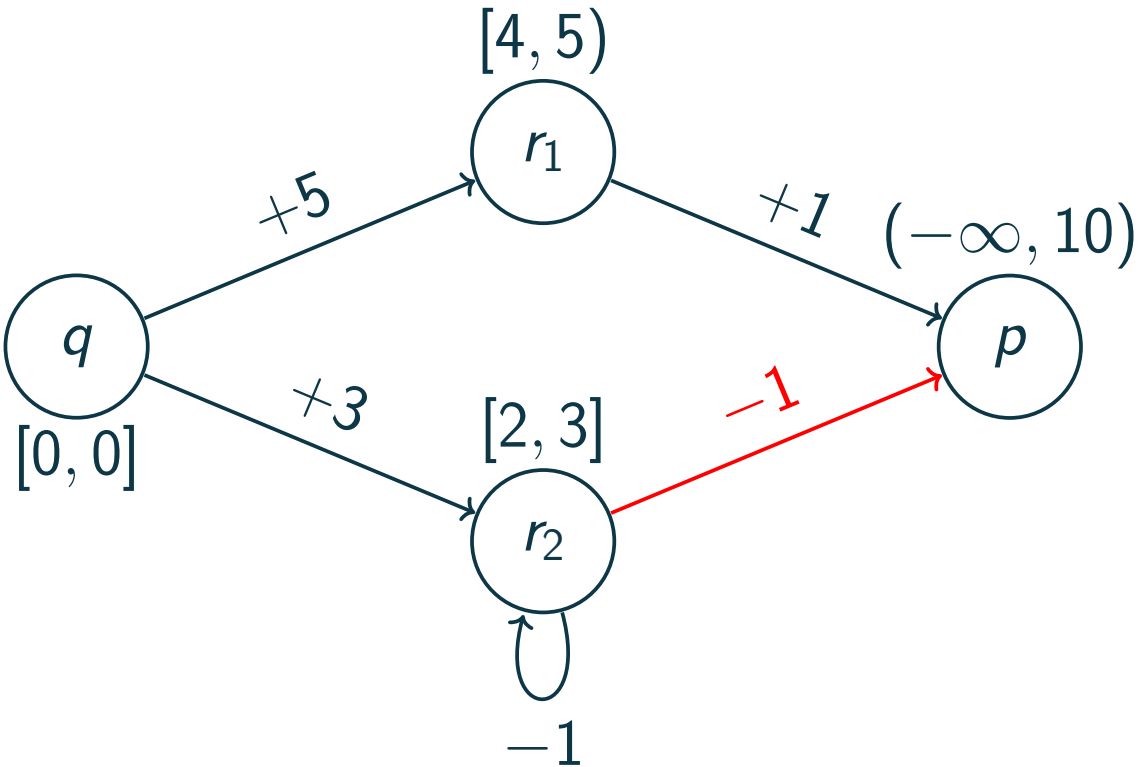
Run: $q(0) \to r_1(5) \to r_2(3)$

One-Counter Automata (OCA)



Run: $q(0) \not\to r_1(5) \to r_2(3) \to r_2(2)$

One-Counter Automata (OCA)

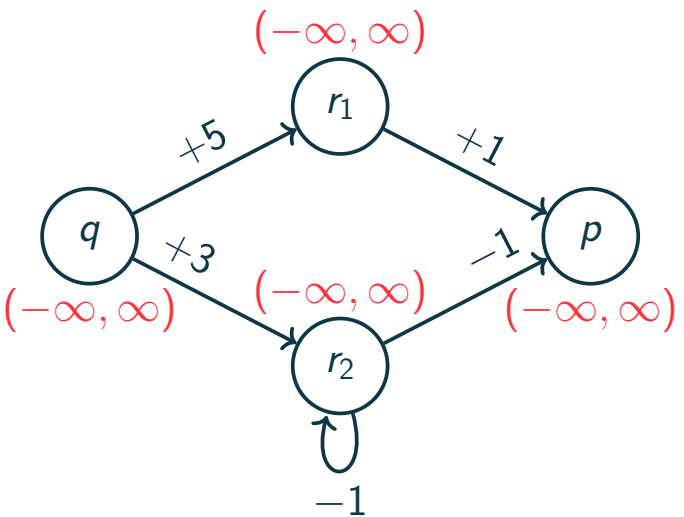


$\rightarrow r_2(3) \rightarrow r_2(2) \rightarrow p(1)$

Run: $q(0) \not\rightarrow r_1(5)$

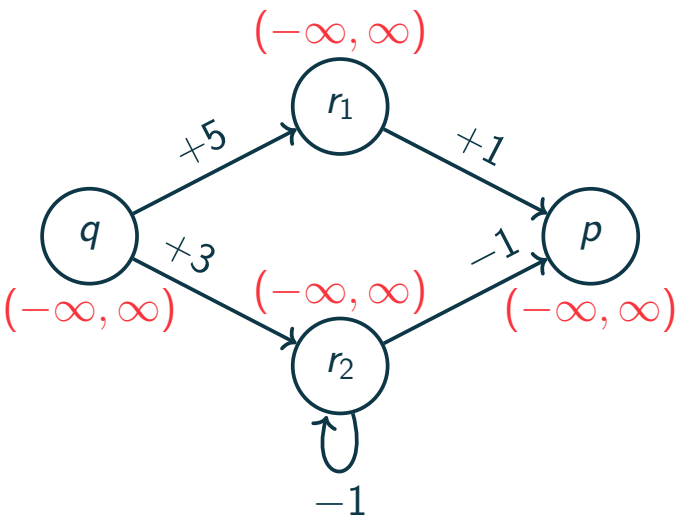
Variants of One-Counter Automata (OCA)

Guardless OCA

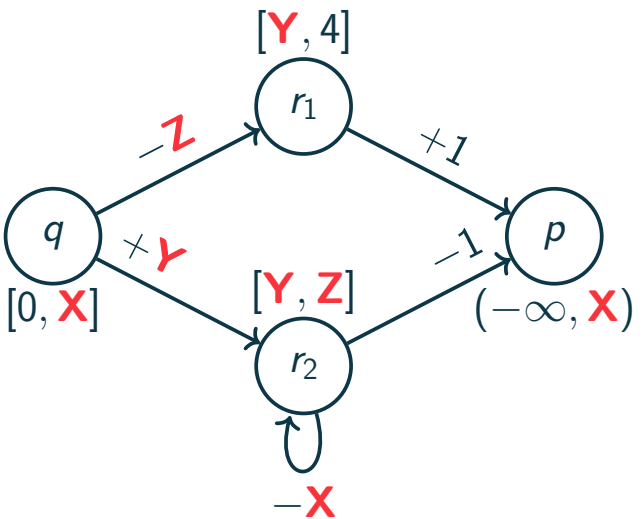


Variants of One-Counter Automata (OCA)

Guardless OCA



Parametric OCA



Reachability in One-Counter Automata (OCA)

What is the state-of-the-art?

Guardless OCA:
NP-complete
[Haase et al.,'09]

Reachability in One-Counter Automata (OCA)

What is the state-of-the-art?

Guardless OCA:
NP-complete
[Haase et al.,'09]

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

Reachability in One-Counter Automata (OCA)

What is the state-of-the-art?

Guardless OCA:
NP-complete
[Haase et al.,'09]

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

Parametric OCA:
Decidability unknown

Reachability in One-Counter Automata (OCA)

What is the state-of-the-art?

Guardless OCA:
NP-complete
[Haase et al.,'09]

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

Parametric OCA:
Decidability unknown

Our goal:
Overapproximate OCA efficiently!

Contributions

Propose novel model:

Continuous One-Counter Automata (COCA)

Overapproximation for OCA with tractable complexity

Contributions

Propose novel model:

Continuous One-Counter Automata (COCA)

Overapproximation for OCA with tractable complexity

Prove complexity results:

Contributions

Propose novel model:

Continuous One-Counter Automata (COCA)

Overapproximation for OCA with tractable complexity

Prove complexity results:

Reachability in **guardless COCA**:
In $\mathbf{NC}^2$

Contributions

Propose novel model:

Continuous One-Counter Automata (COCA)

Overapproximation for OCA with tractable complexity

Prove complexity results:

Reachability in **guardless COCA**:
In **NC**^{2}

Reachability in **standard COCA**:
In **P-time**

Contributions

Propose novel model:
Continuous One-Counter Automata (COCA)
Overapproximation for OCA with tractable complexity

Prove complexity results:

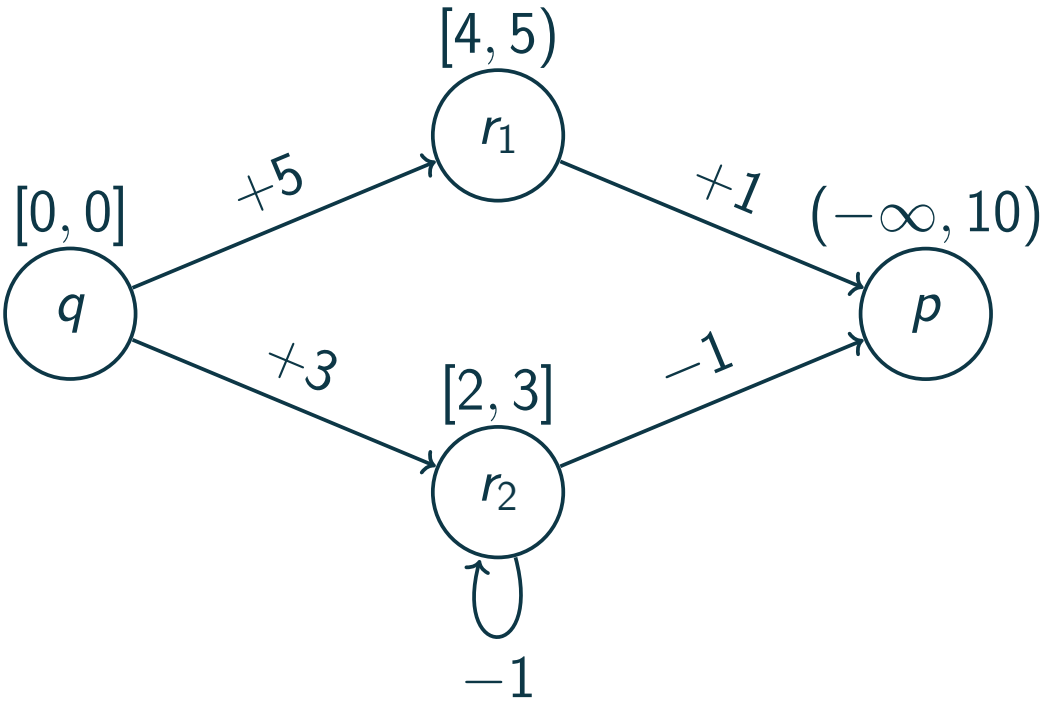
Reachability in **guardless COCA**:
In **NC^2**

Reachability in **standard COCA**:
In **P-time**

Reachability in **parametric COCA**:
NP-complete

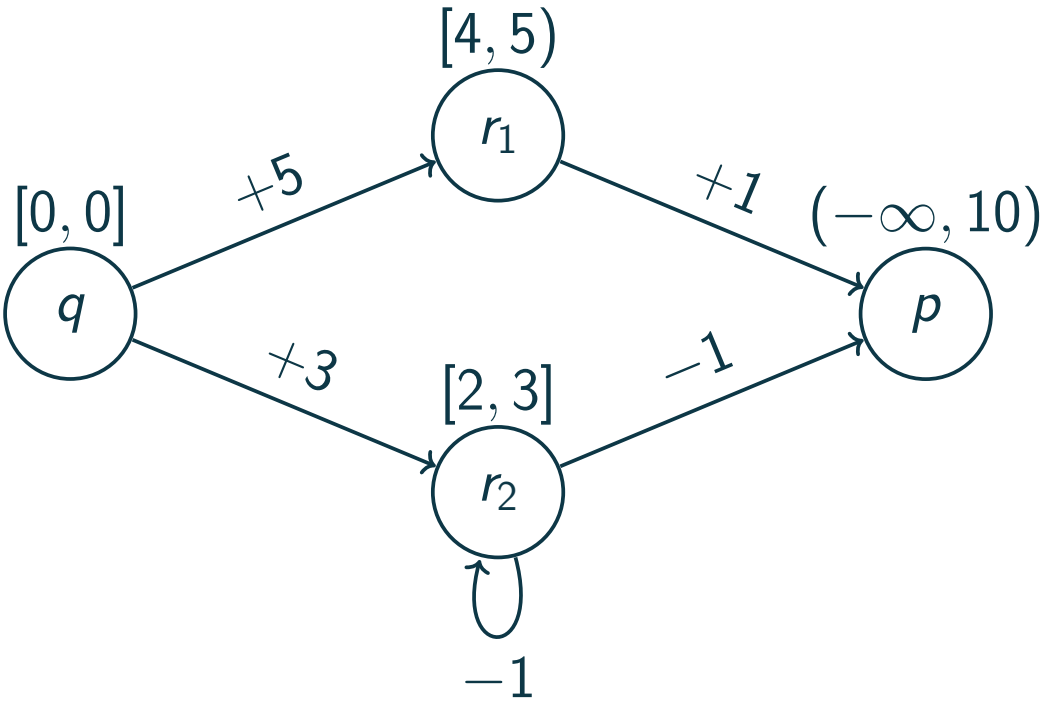
Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



Overapproximating One-Counter Automata

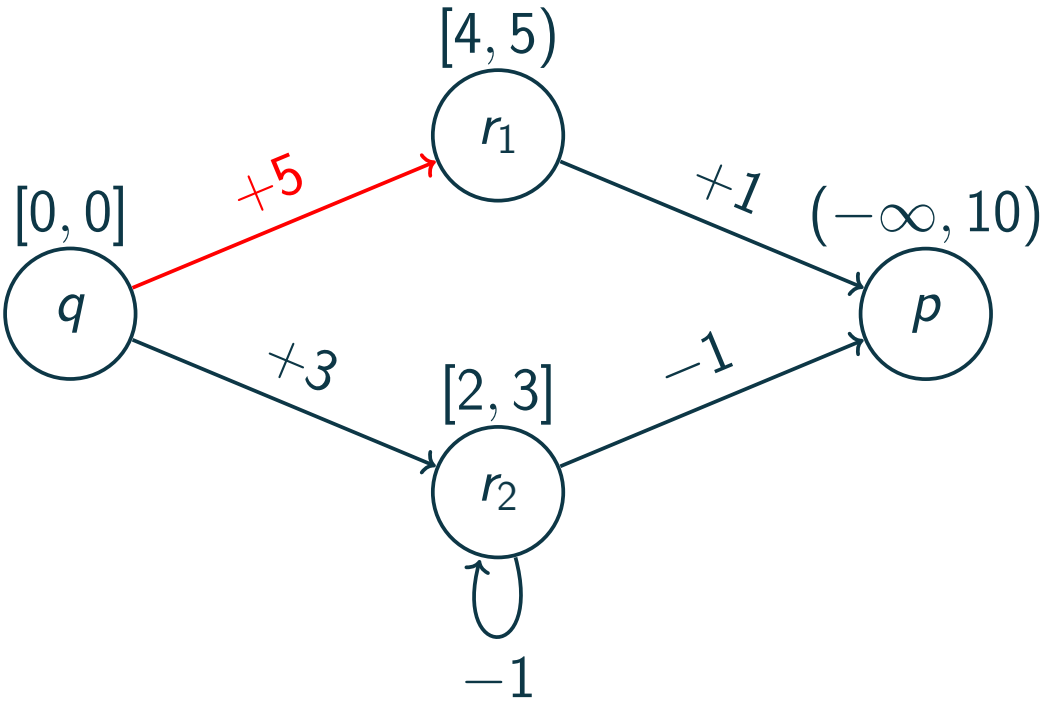
Continuous One-Counter Automata (COCA)



Run: $q(0)$

Overapproximating One-Counter Automata

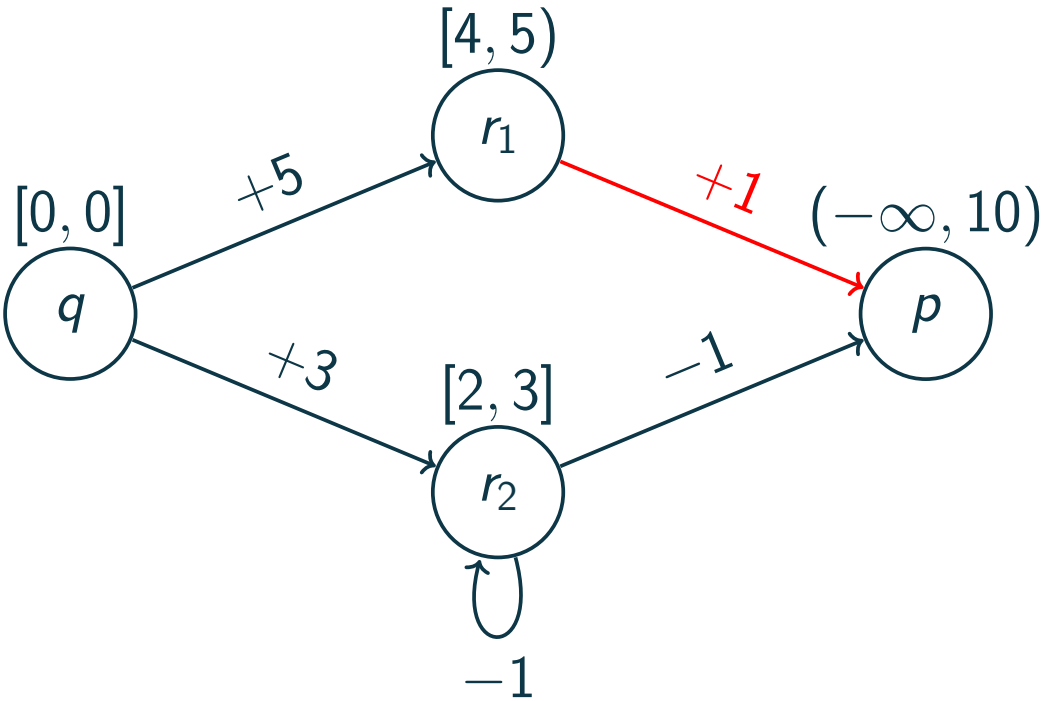
Continuous One-Counter Automata (COCA)



Run: $q(0) \xrightarrow{4/5} r_1(4)$

Overapproximating One-Counter Automata

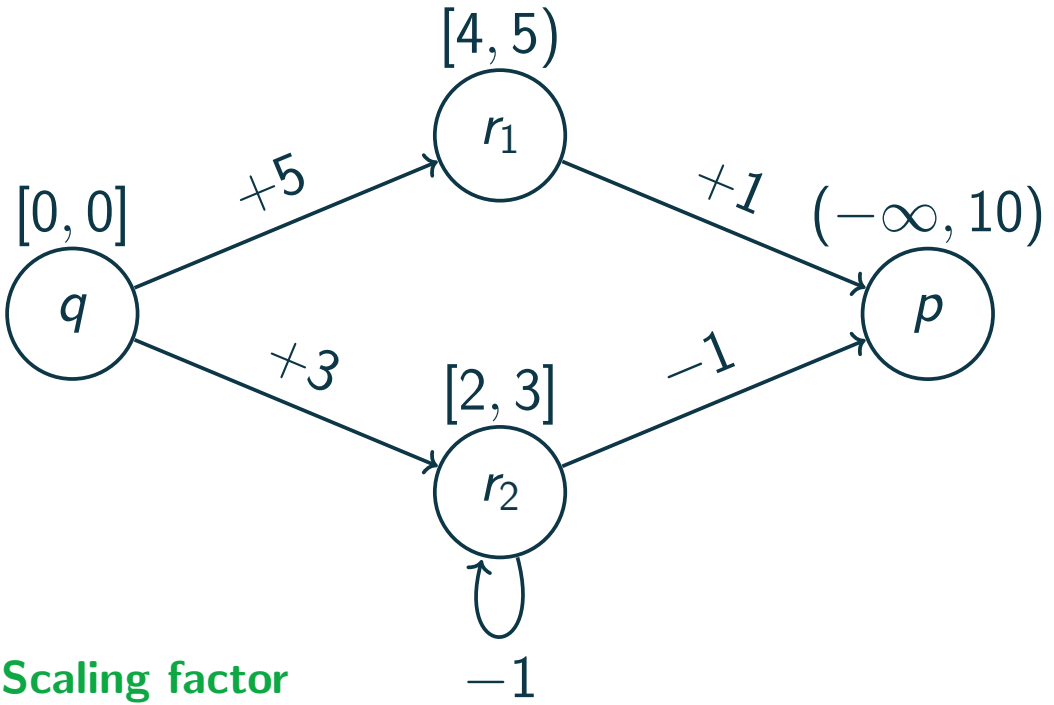
Continuous One-Counter Automata (COCA)



Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



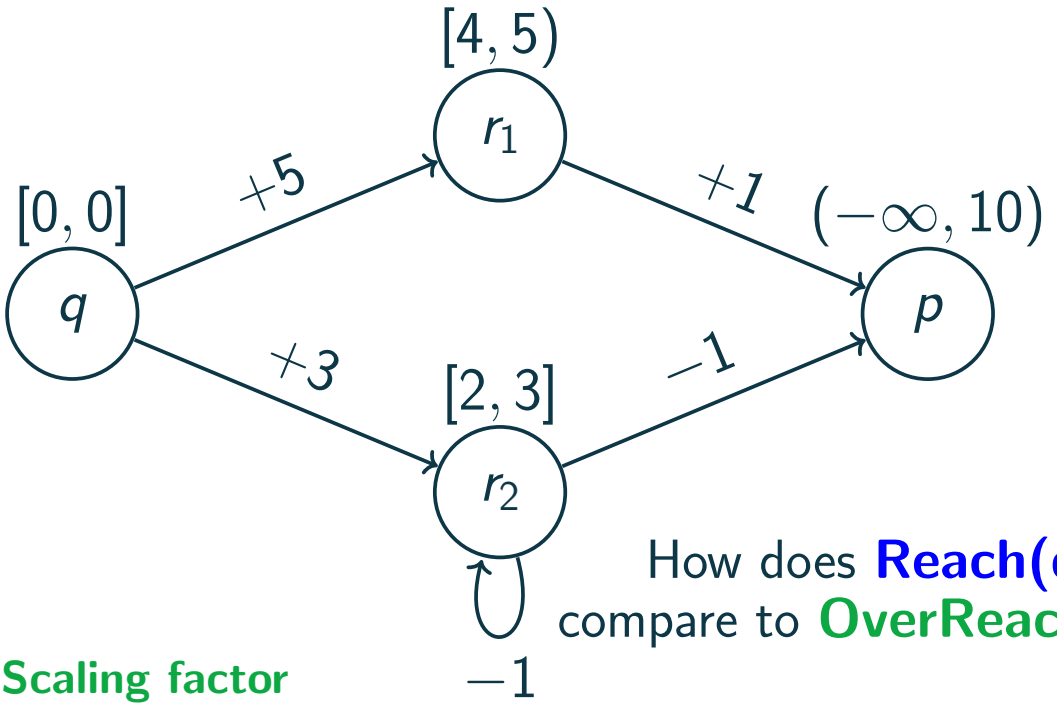
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

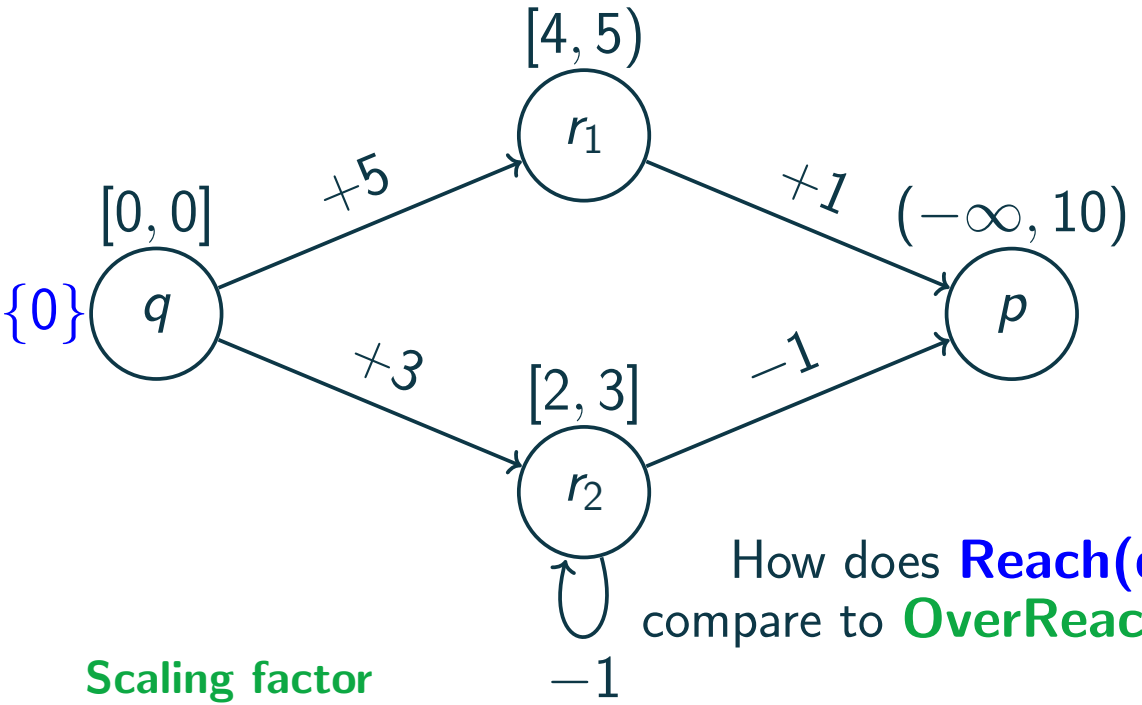
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

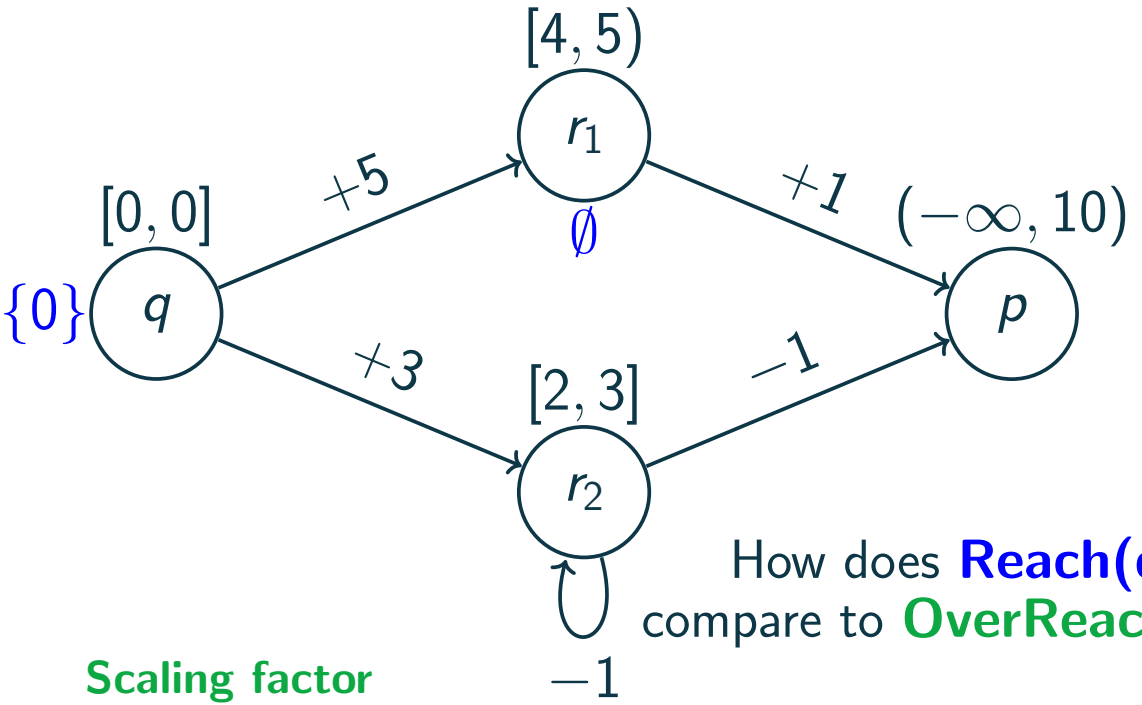
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

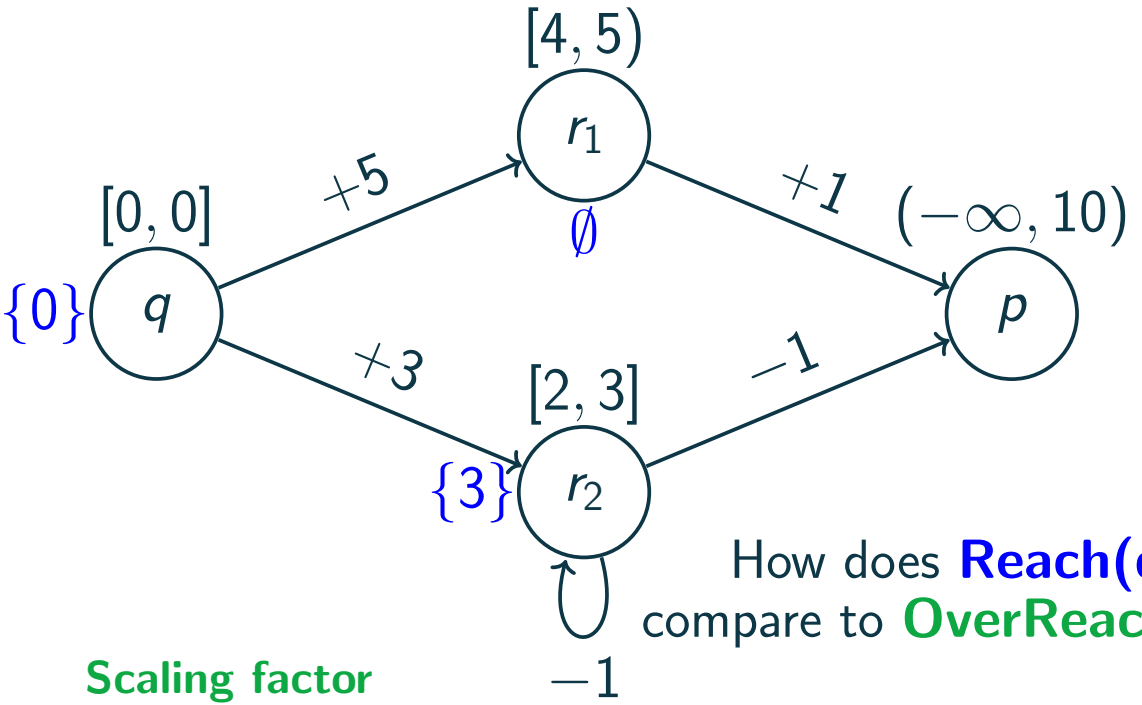
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

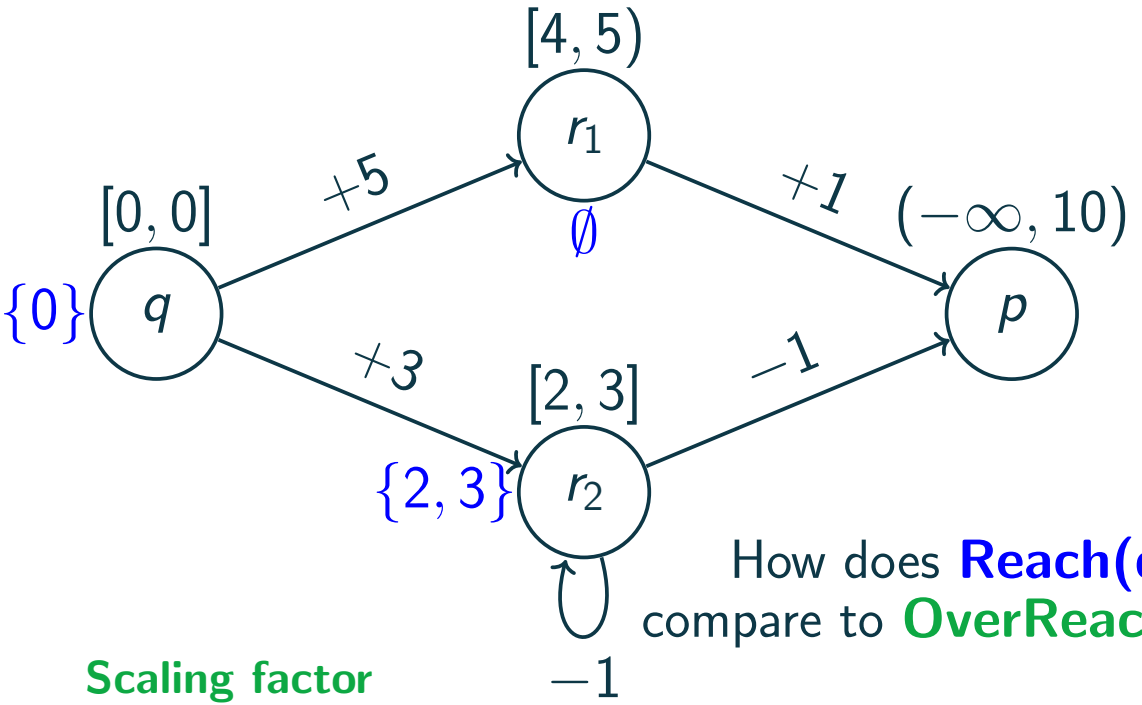
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

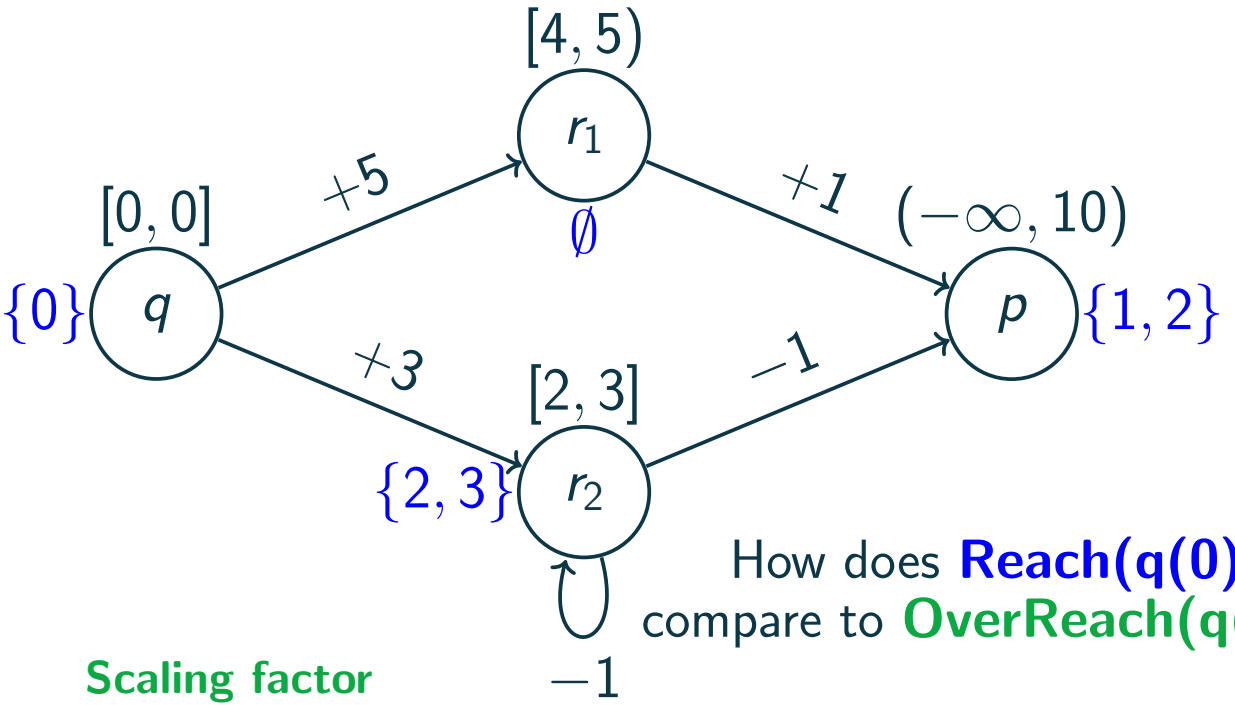
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

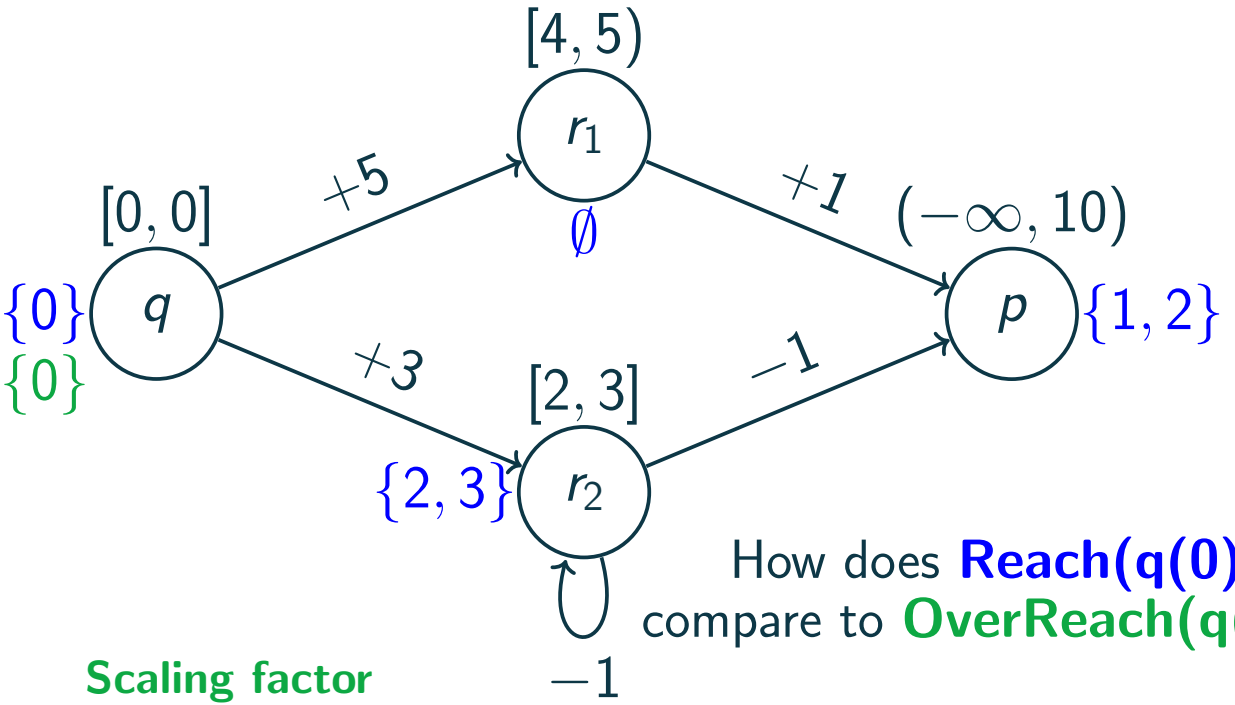
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

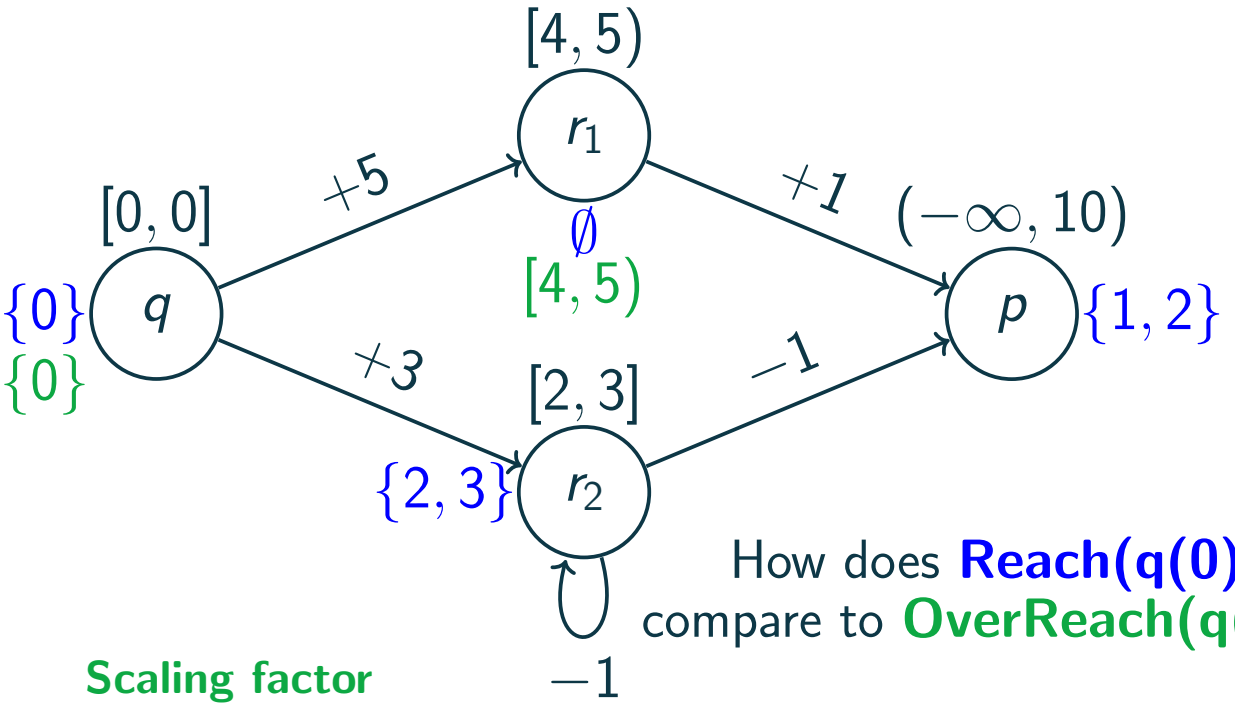
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

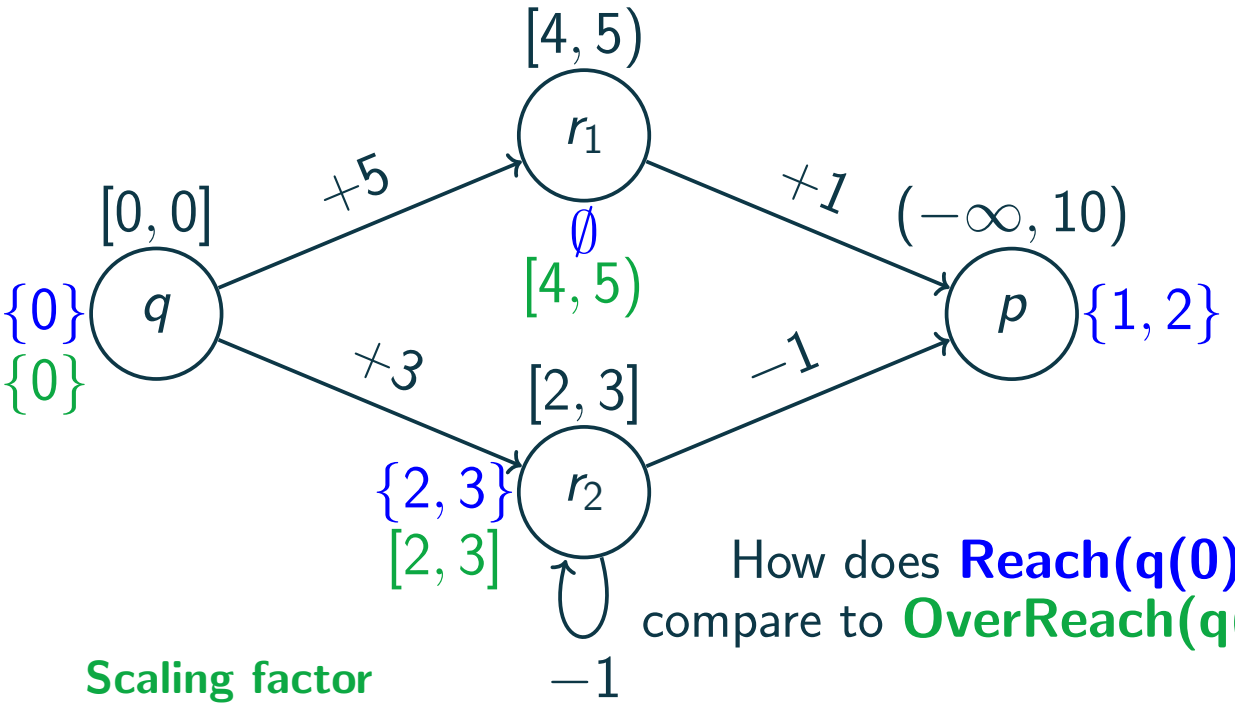
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

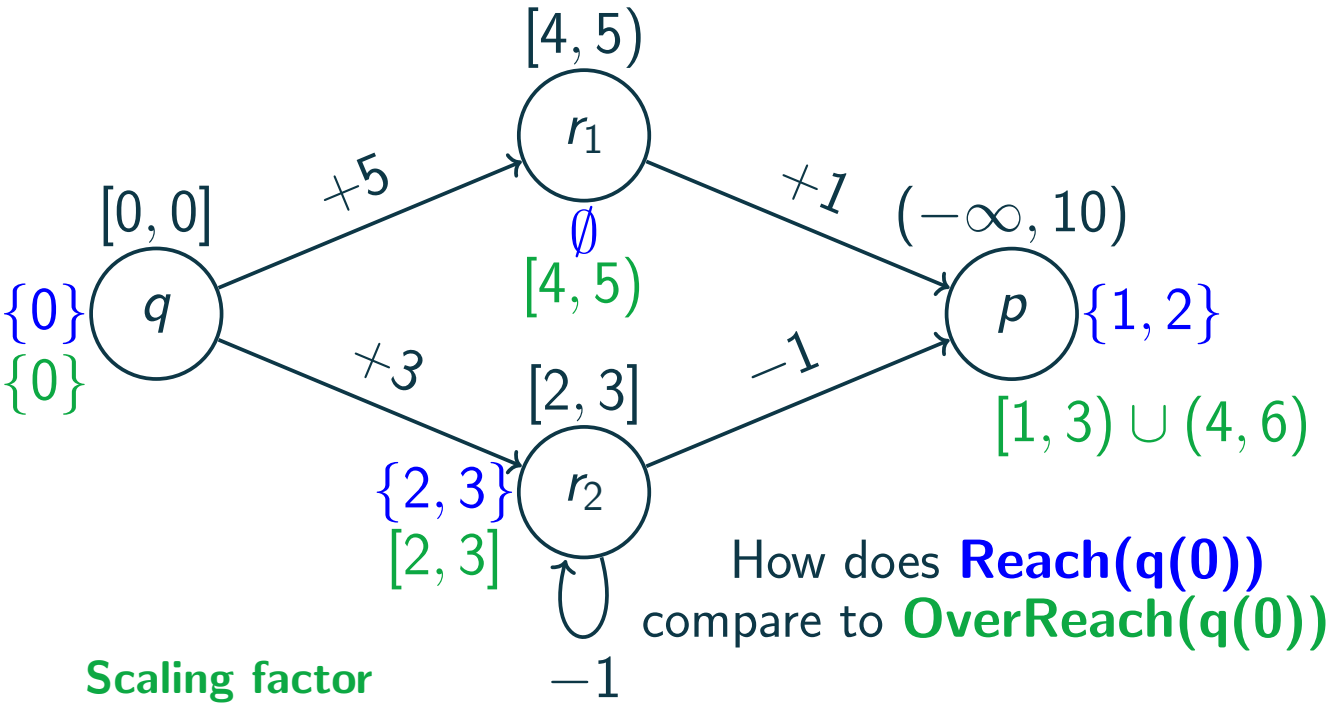
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

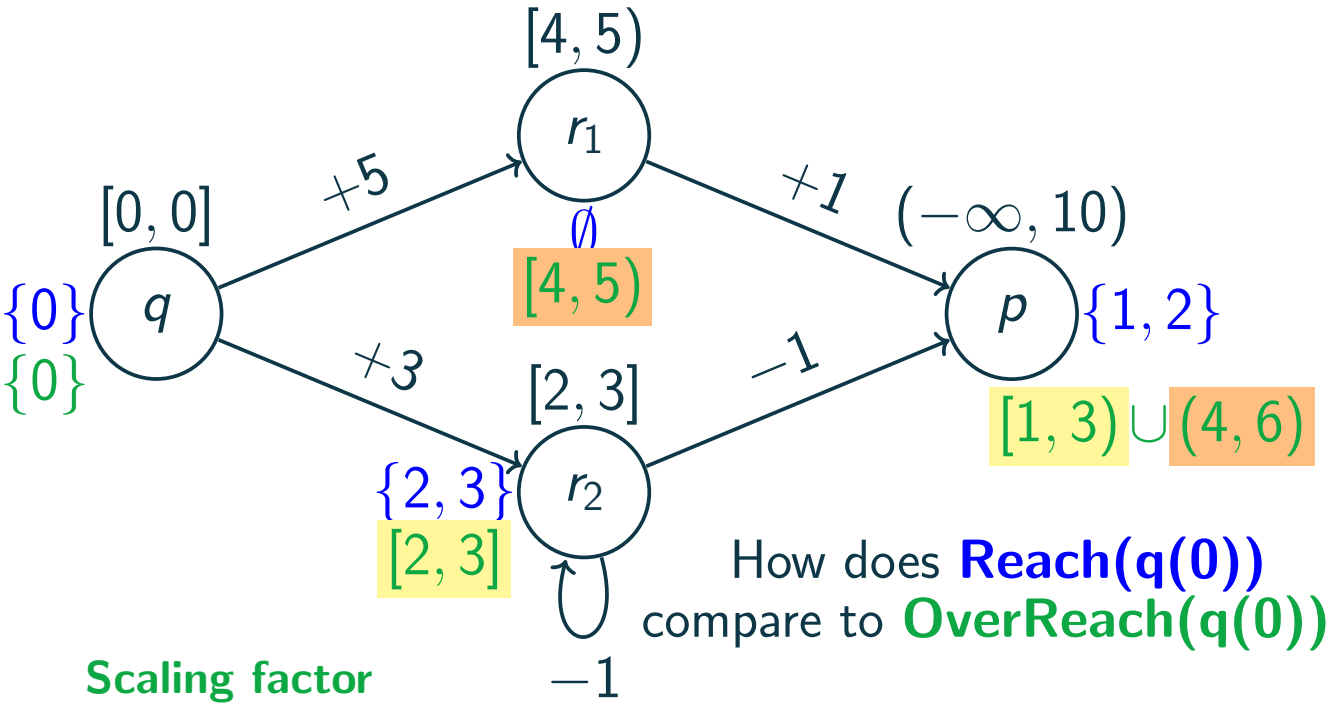
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

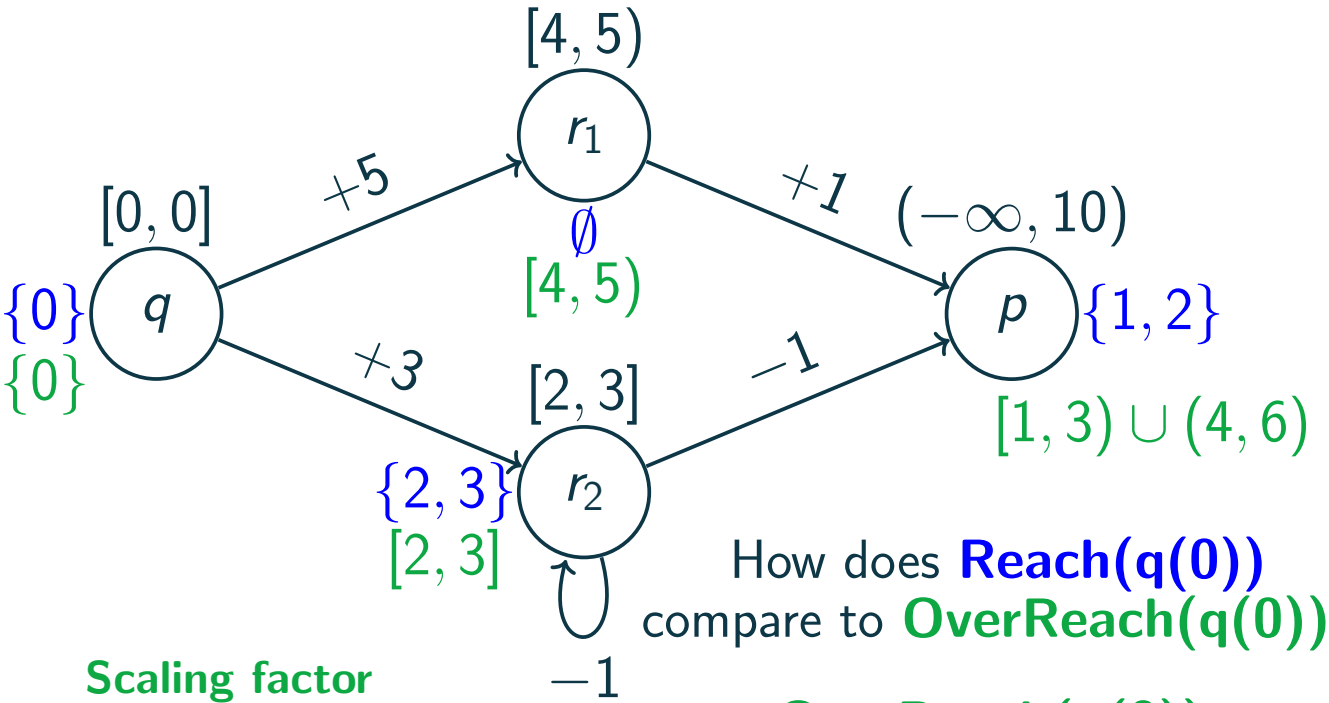
Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Overapproximating One-Counter Automata

Continuous One-Counter Automata (COCA)



How does **Reach(q(0))** compare to **OverReach(q(0))**?

OverReach(q(0)): union of (few) intervals

Scaling factor

$\beta \in (0, 1]$

Run: $q(0) \xrightarrow{4/5} r_1(4) \xrightarrow{1} p(5)$

Reachability in Continuous OCA (COCA)

...has much lower complexity

Guardless OCA:
NP-complete
[Haase et al.,'09]

$\Rightarrow$

Guardless COCA:
in **NC^2** (below P-time)

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

Parametric OCA:
Decidability unknown

Reachability in Continuous OCA (COCA)

...has much lower complexity

Guardless OCA:
NP-complete
[Haase et al.,'09]

$\Rightarrow$

Guardless COCA:
in **NC^2** (below P-time)

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

Parametric OCA:
Decidability unknown

Reachability in Continuous OCA (COCA)

...has much lower complexity

Guardless OCA:
NP-complete
[Haase et al.,'09]

$\Rightarrow$

Guardless COCA:
in **NC^2** (below P-time)

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

NC^2: **Polynomially** many **random-access machines** running for at most **$O(\log^2 n)$** steps in parallel

Parametric OCA:
Decidability unknown

Reachability in Continuous OCA (COCA)

...has much lower complexity

Guardless OCA:
NP-complete
[Haase et al.,'09]

$\Rightarrow$

Guardless COCA:
in $\mathbf{NC^2}$ (below P-time)

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

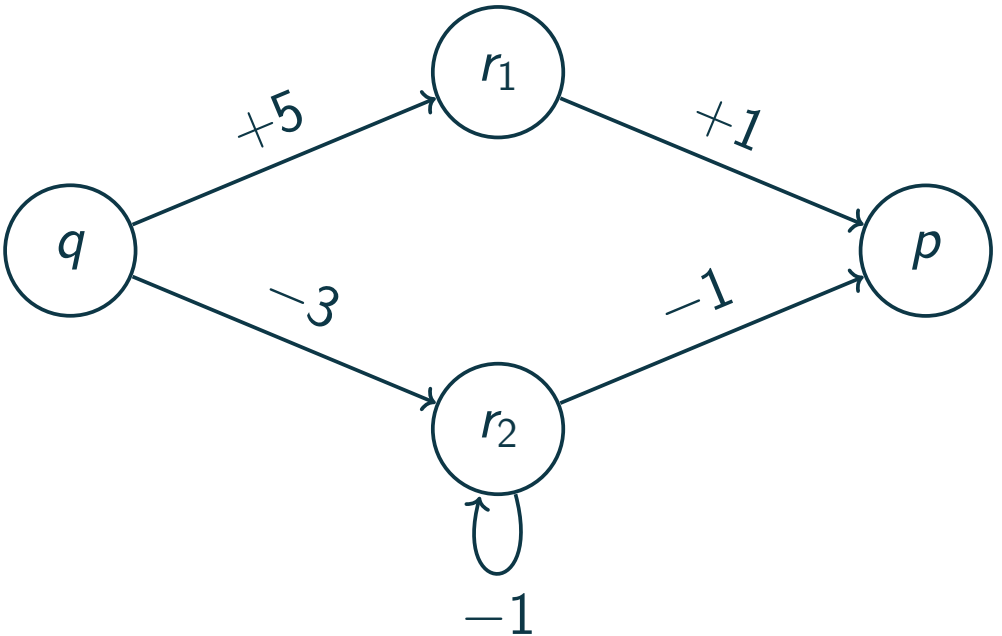
$\mathbf{NC^2}$: **Polynomially** many **random-access machines** running for at most $\mathbf{O(\log^2 n)}$ steps in parallel

Parametric OCA:
Decidability unknown

Notably:
Graph reachability $\in \mathbf{NC^2}$
Also for weighted graphs!

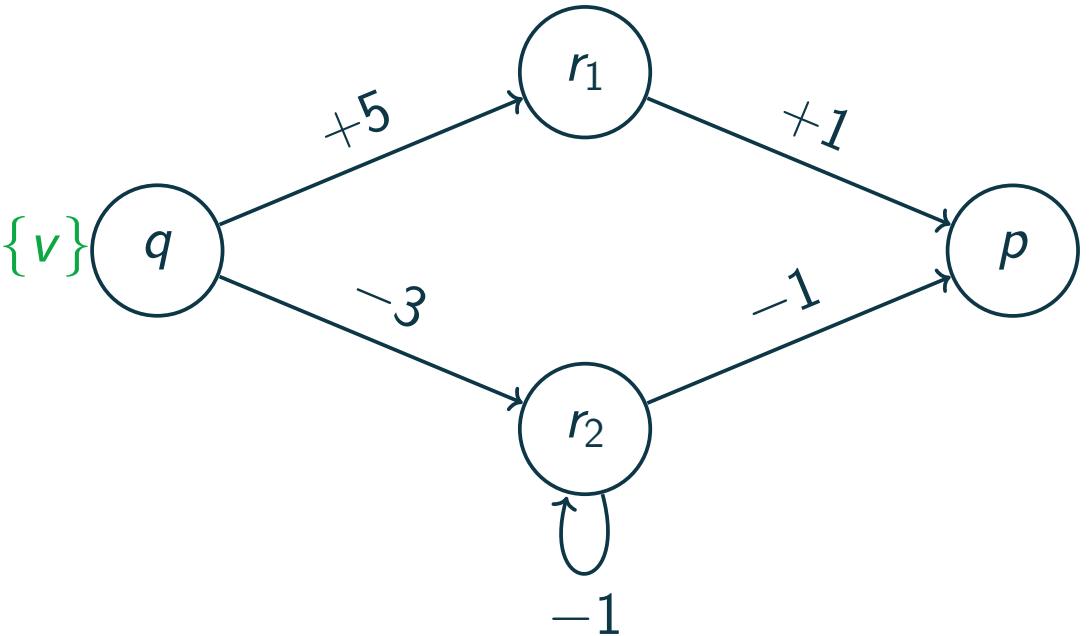
Guardless COCA

OverReach is a single interval (with a gap)



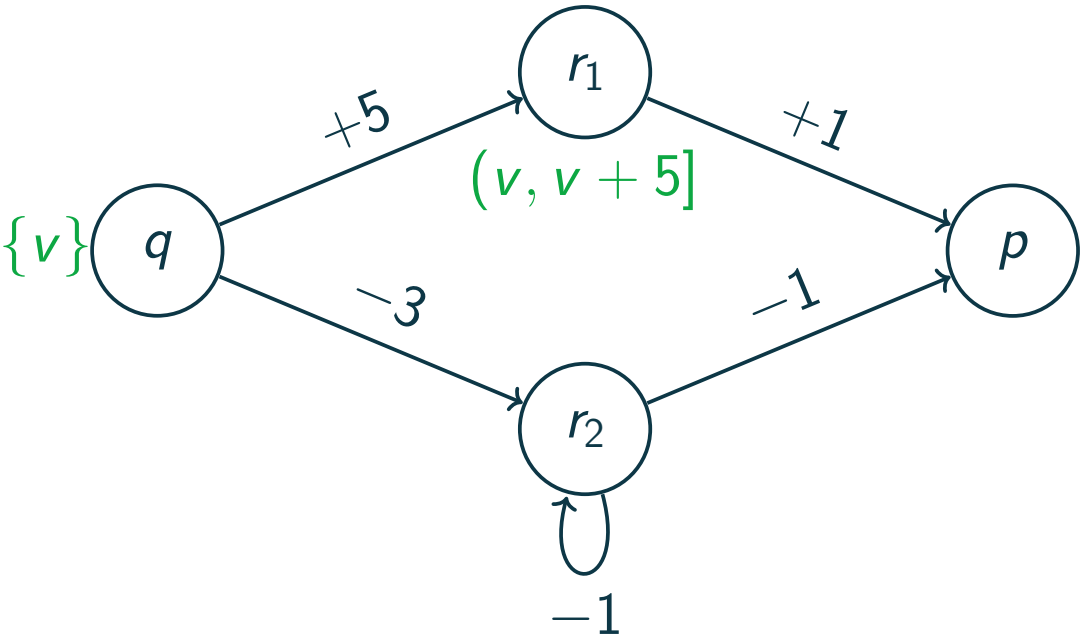
Guardless COCA

OverReach is a single interval (with a gap)



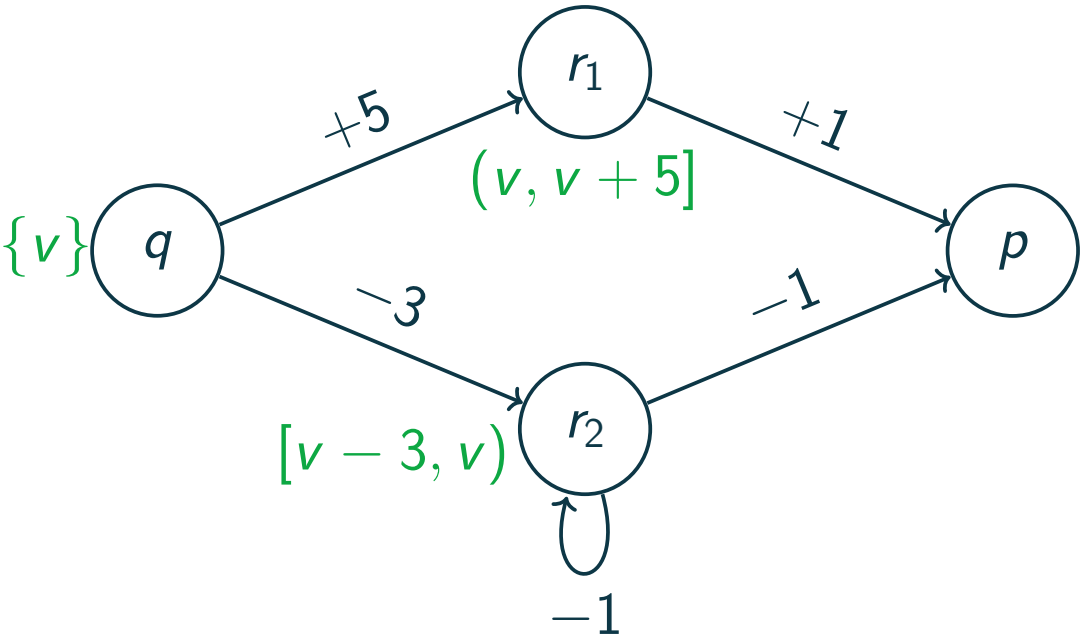
Guardless COCA

OverReach is a single interval (with a gap)



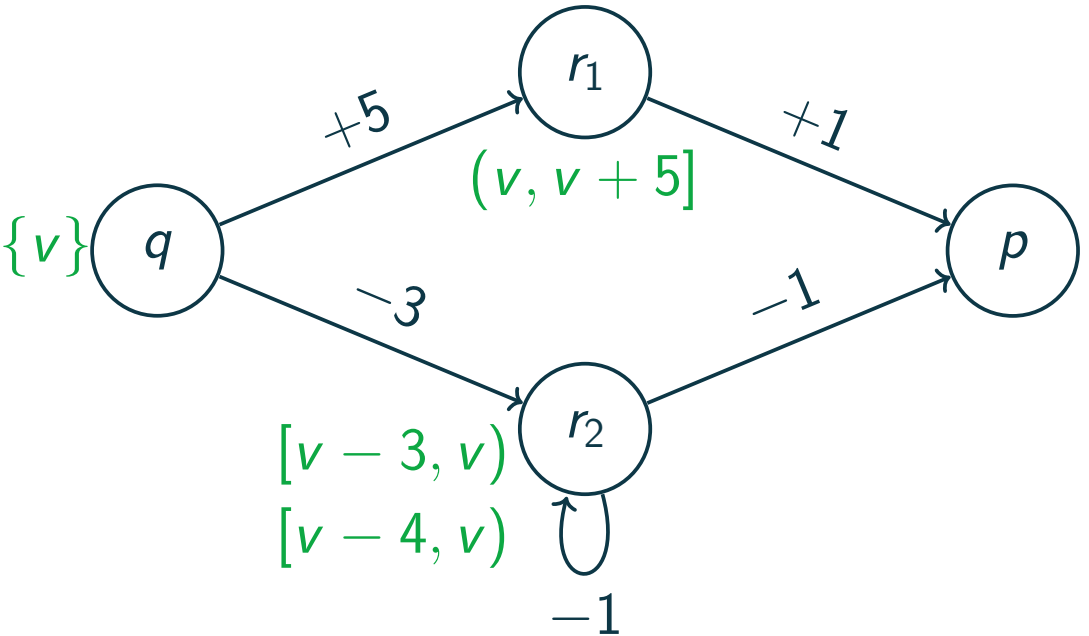
Guardless COCA

OverReach is a single interval (with a gap)



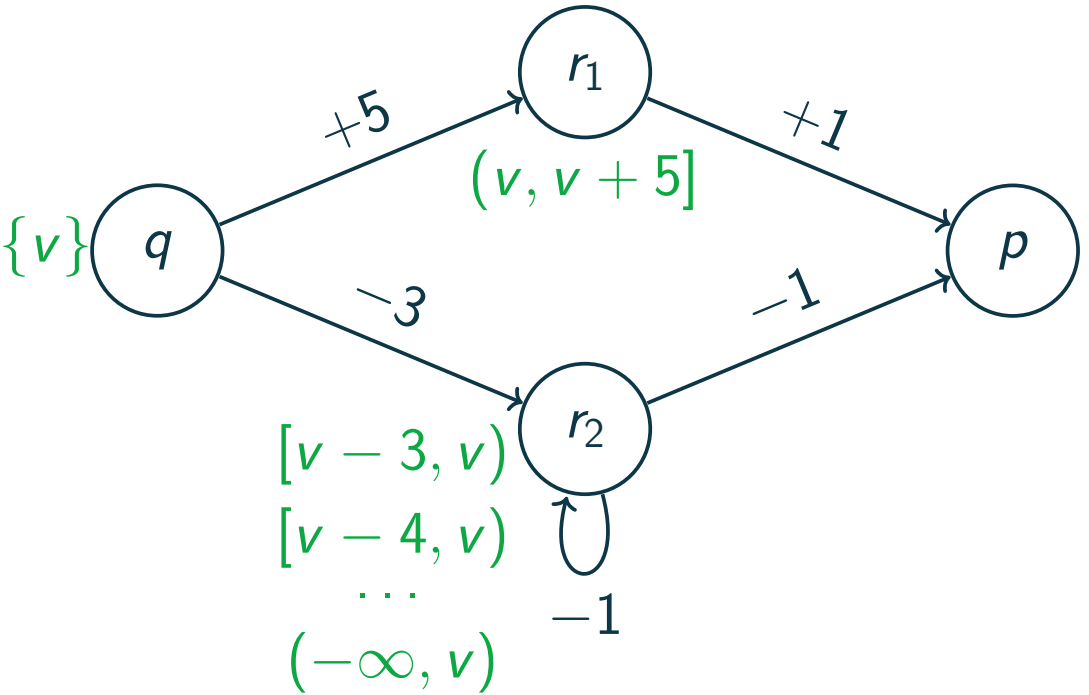
Guardless COCA

OverReach is a single interval (with a gap)



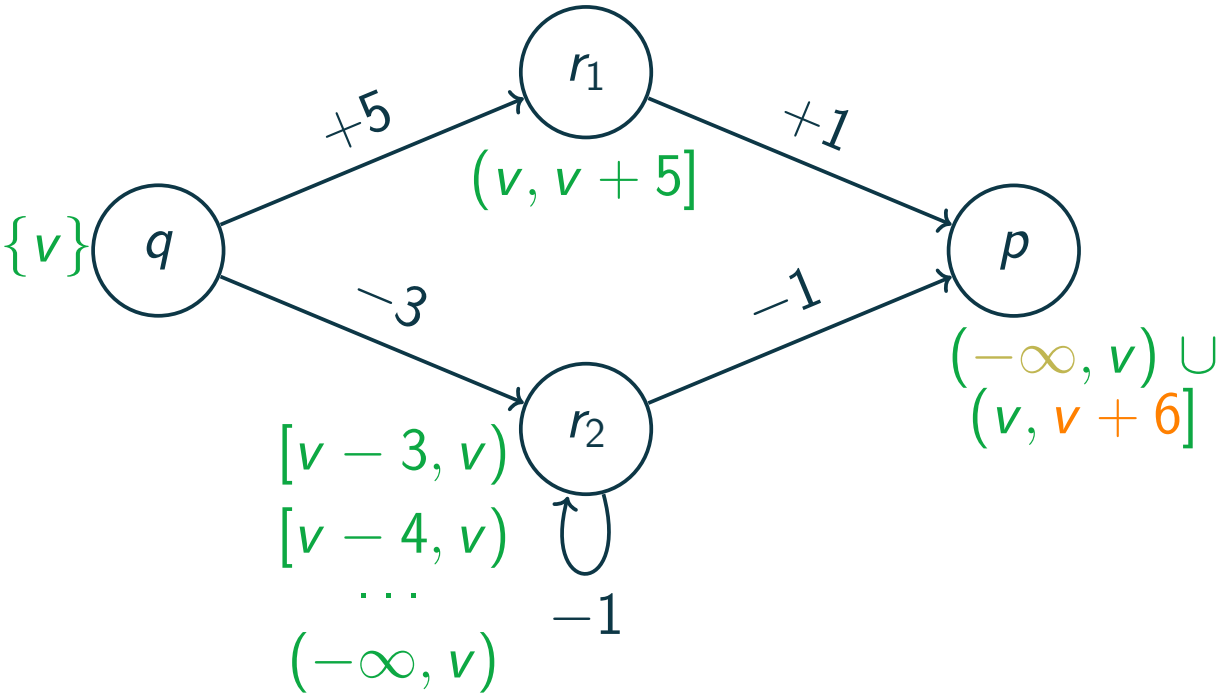
Guardless COCA

OverReach is a single interval (with a gap)



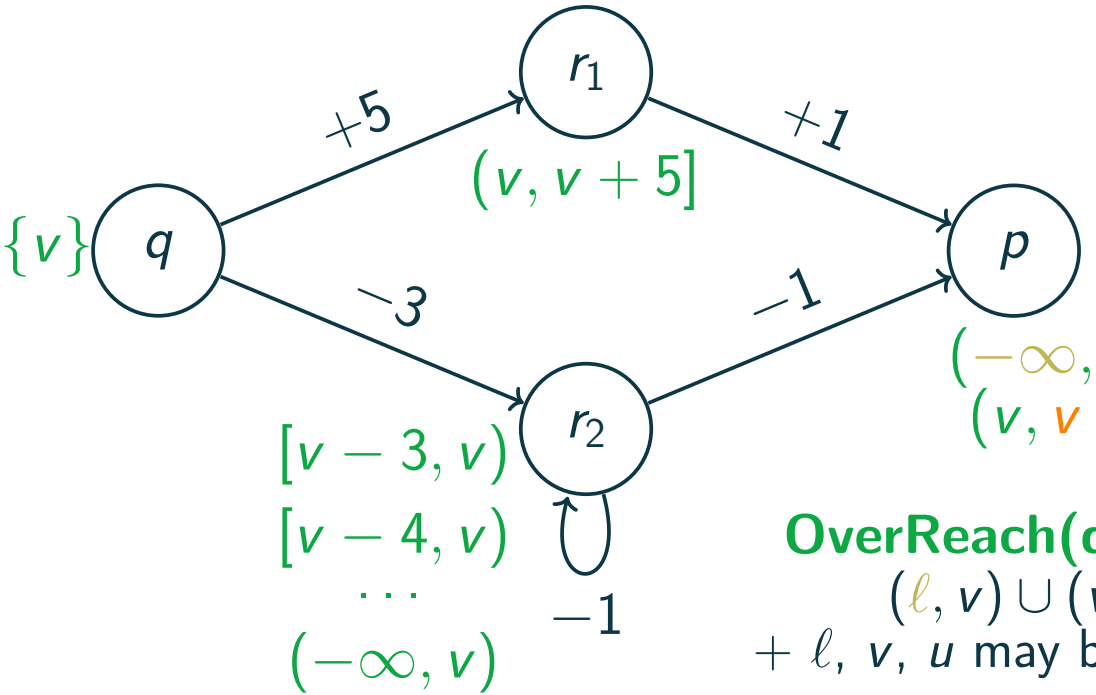
Guardless COCA

OverReach is a single interval (with a gap)



Guardless COCA

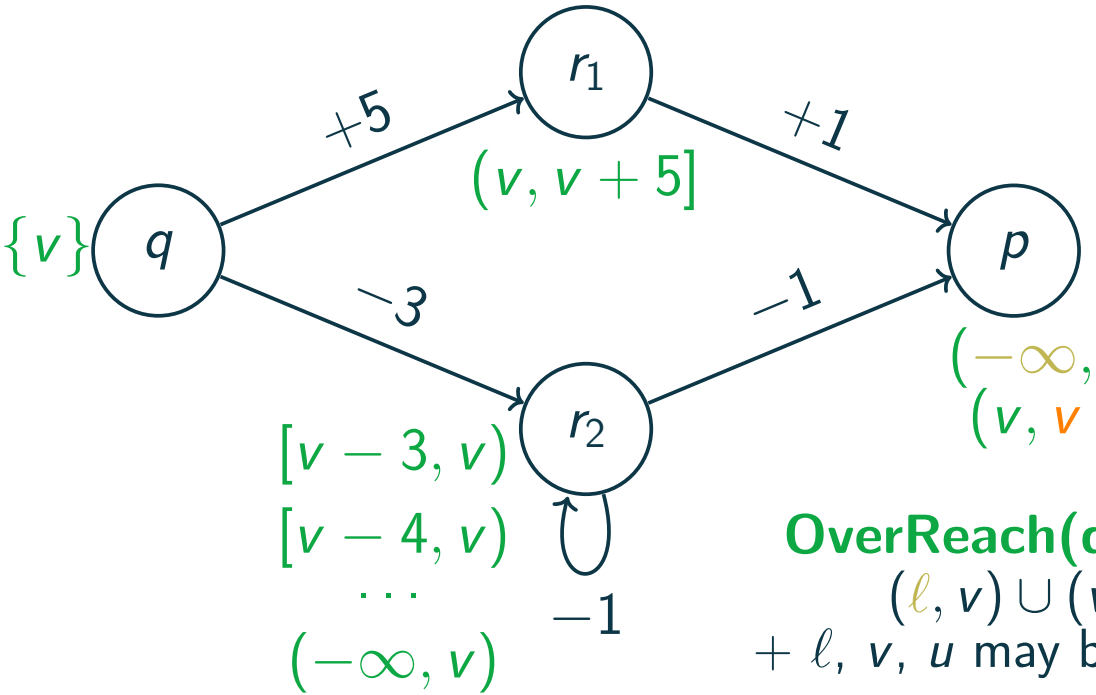
OverReach is a single interval (with a gap)



OverReach(q(v))[p]:
$(\ell, v) \cup (v, u)$
$+ \ell$, v, u may be included

Guardless COCA

OverReach is a single interval (with a gap)



OverReach(q(v))[p]:
$(\ell, v) \cup (v, u)$
$+ \ell$, v, u may be included

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

how do we compute this? ⇢ OverReach(p(v))[q]: $(\ell, v) \cup (v, u)$ ⇠ symmetric!

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

how do we compute this? ⇢ OverReach(p(v))[q]: $(\ell, v) \cup (v, u)$ ⇠ symmetric!

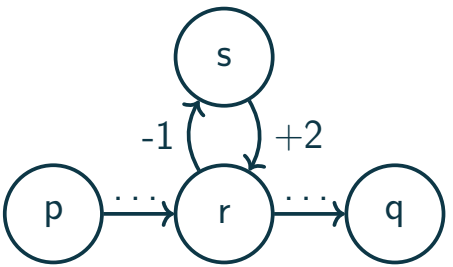
Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

how do we compute this? ⇢ OverReach(p(v))[q]: $(\ell, v) \cup (v, u)$ ⇠ symmetric!

Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$



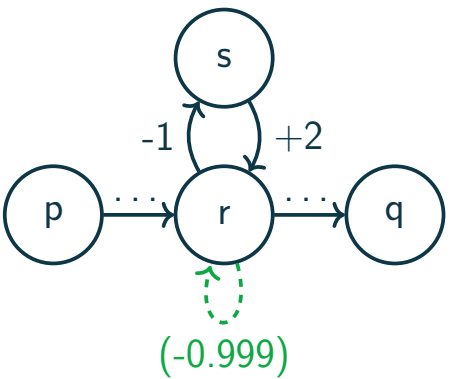
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

how do we compute this? ⇢ $(\ell, v) \cup (v, u)$ ⇠ symmetric!

OverReach(p(v))[q]:

Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$



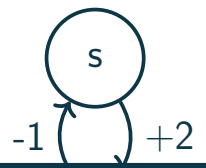
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

how do we compute this? ⇢ $(\ell, v) \cup (v, u)$ ⇠ symmetric!

OverReach(p(v))[q]:

Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$



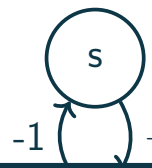
Check for each node n:
Is there a path from n to n with a negative edge?
$\Rightarrow \in \mathsf{NC}^2$

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

how do we compute this? ⇢ OverReach(p(v))[q]: $(\ell, v) \cup (v, u)$ ⇠ symmetric!

Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$



Check for each node n:
Is there a path from n to n with a negative edge?
$\Rightarrow \in \mathsf{NC}^2$

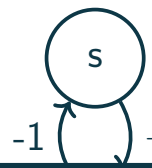
Otherwise: $\ell = v -$ **min. sum** of **negative edges** among **paths** $p \to q$

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

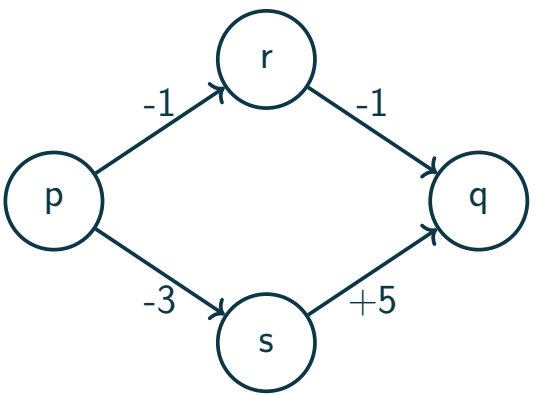
how do we compute this? ⇢ OverReach(p(v))[q]: $(\ell, v) \cup (v, u)$ ⇠ symmetric!

Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$



Check for each node n:
Is there a path from n to n with a negative edge?
$\Rightarrow \in \mathsf{NC}^2$

Otherwise: $\ell = v -$ **min. sum** of **negative edges** among **paths** $p \to q$

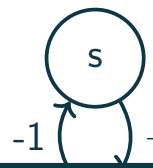


Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

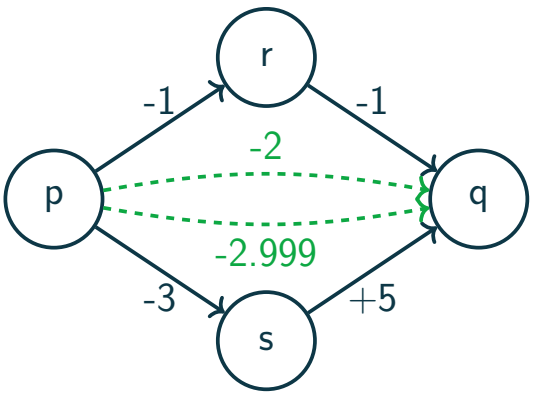
how do we compute this? ⇢ OverReach(p(v))[q]: $(\ell, v) \cup (v, u)$ ⇠ symmetric!

Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$



Check for each node n:
Is there a path from n to n with a negative edge?
$\Rightarrow \in \mathsf{NC}^2$

Otherwise: $\ell = v -$ **min. sum** of **negative edges** among **paths** $p \to q$

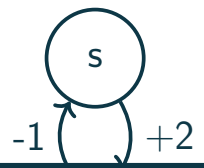


Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

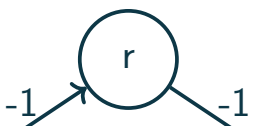
how do we compute this? ⇢ OverReach(p(v))[q]: $(\ell, v) \cup (v, u)$ ⇠ symmetric!

Cycle with a **negative** edge between p and $q \Rightarrow \ell = -\infty$



Check for each node n:
Is there a path from n to n with a negative edge?
$\Rightarrow \in$ NC^2

Otherwise: $\ell = v -$ **min. sum** of **negative edges** among **paths** $p \to q$



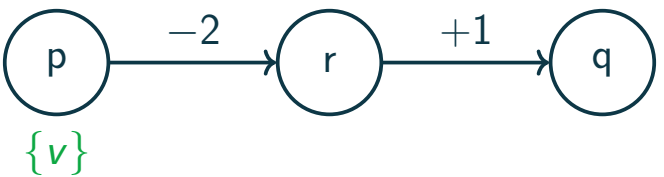
Longest path problem, but:
No cycle with neg. edge
$\Rightarrow \in$ NC^2

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

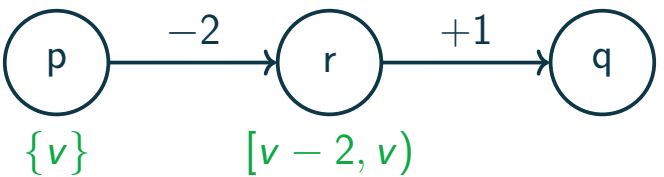
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



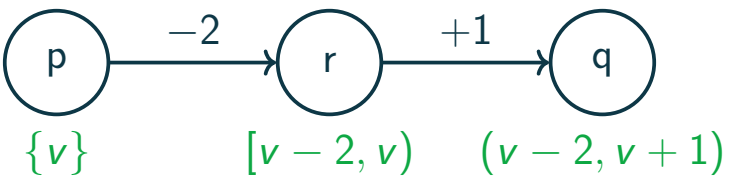
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



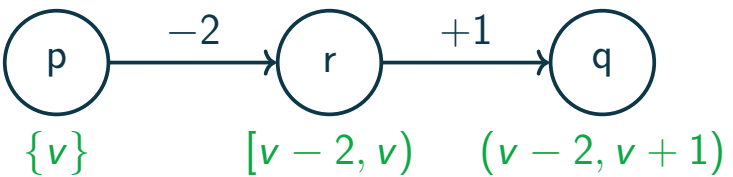
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



Guardless COCA

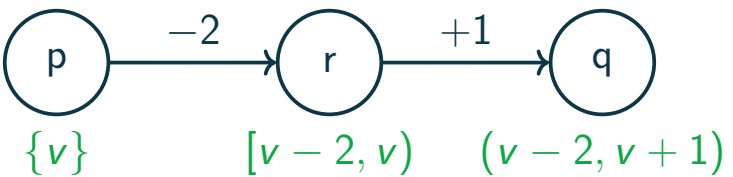
1. Computing ℓ and u 2. Checking membership of ℓ, v, u



v is **included** if **any path** $p \to q$
has positive **and** negative edges

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



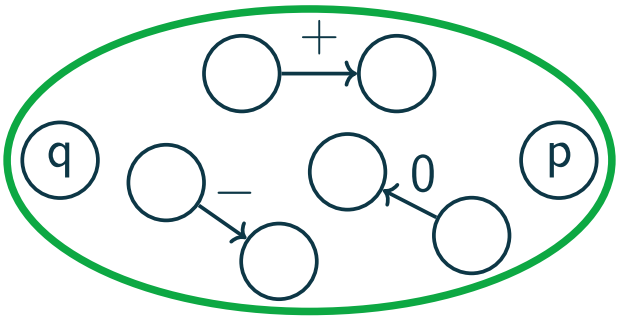
v is **included** if **any path** $p \to q$
has positive **and** negative edges

$\Rightarrow$ Can be checked in $\mathbf{NC^2}$:
Reachability in modified copies of the underlying graph **C**

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

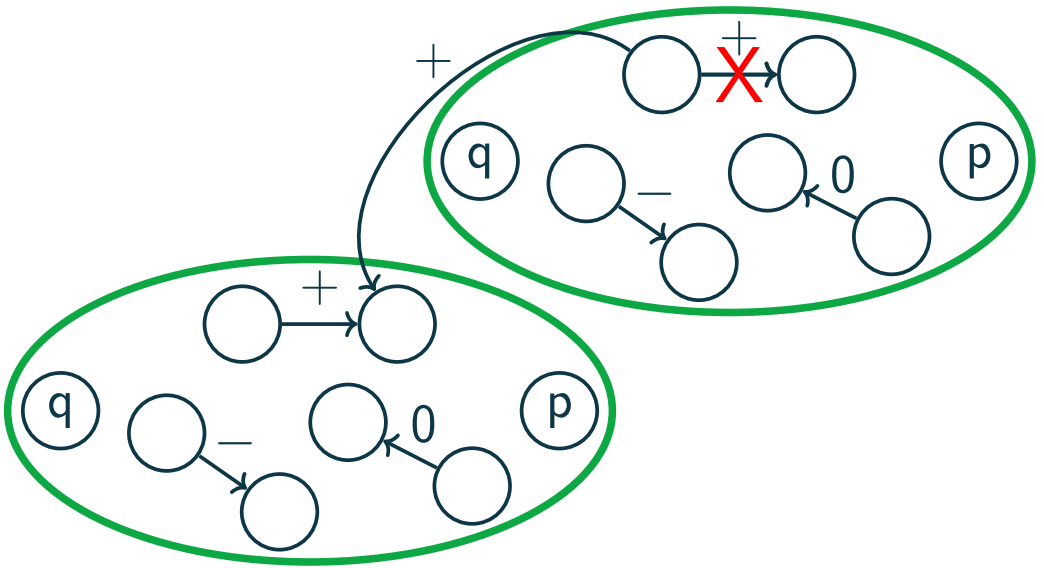
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

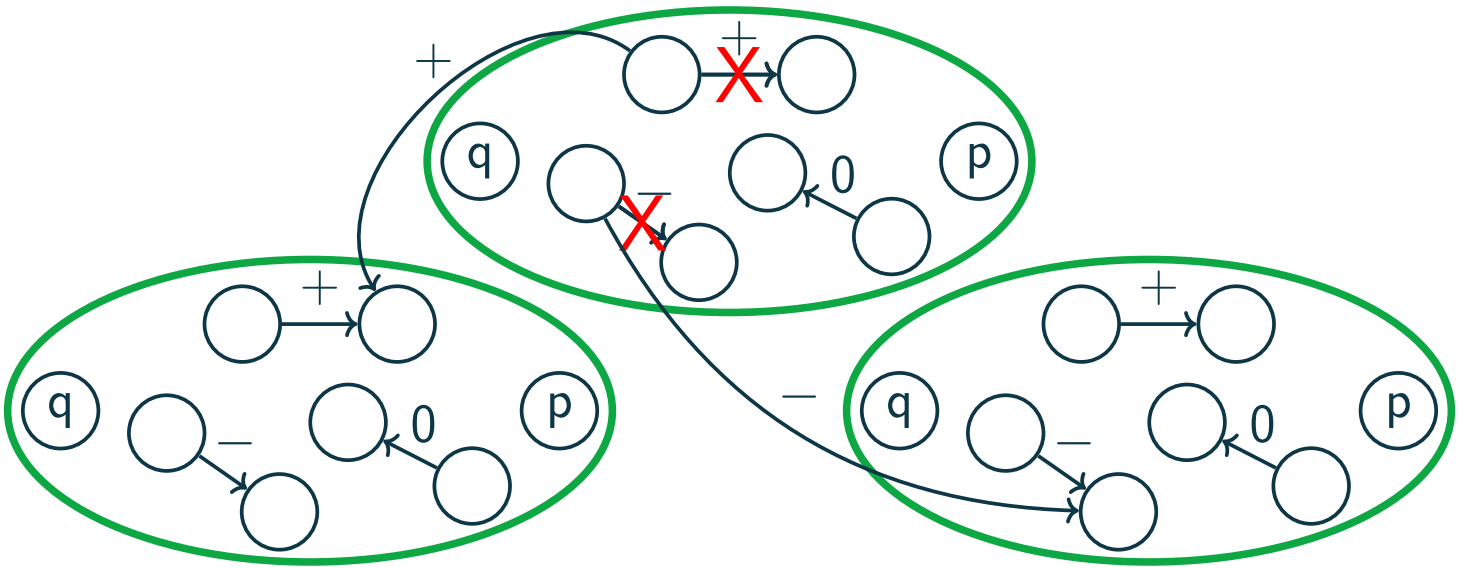
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

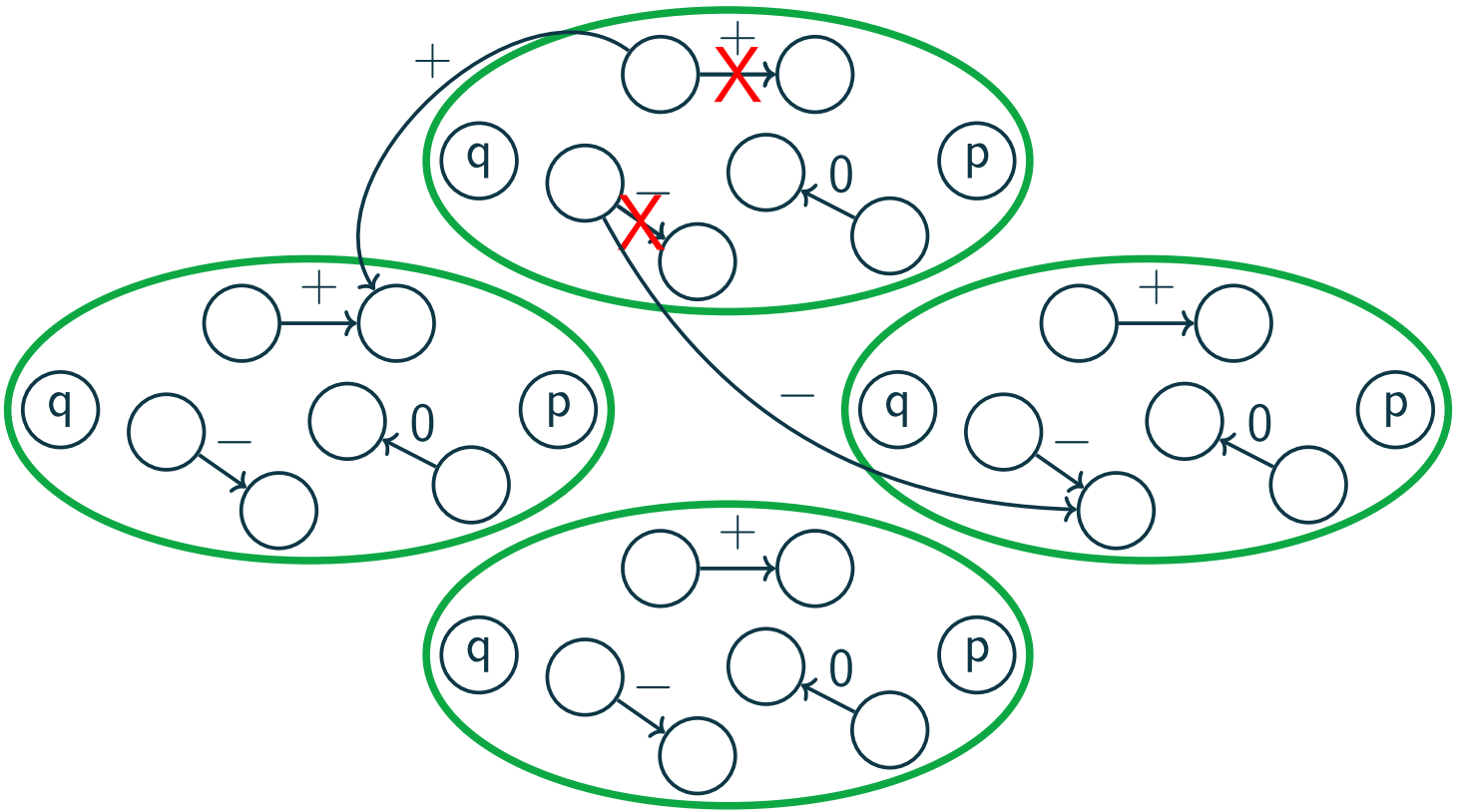
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

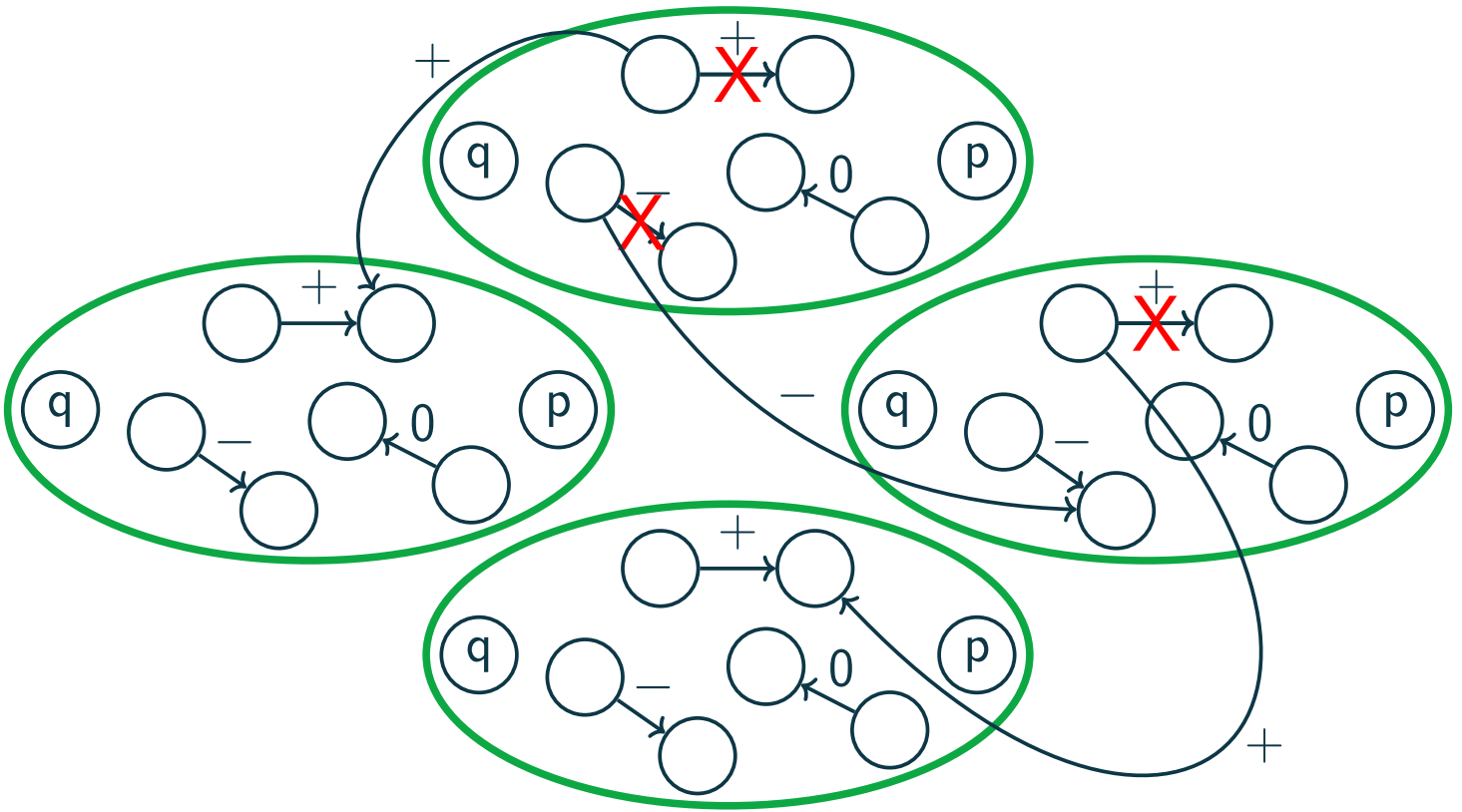
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

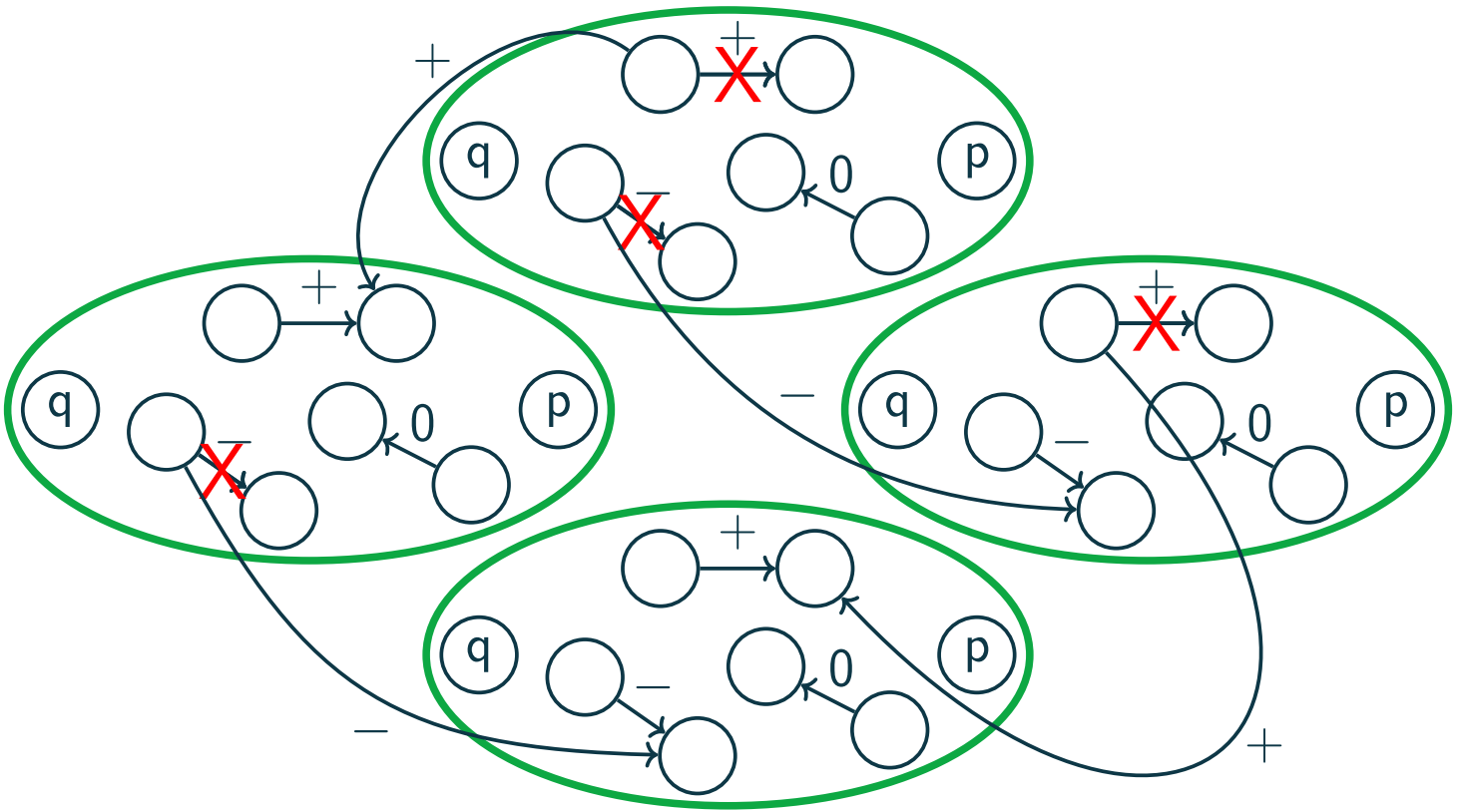
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

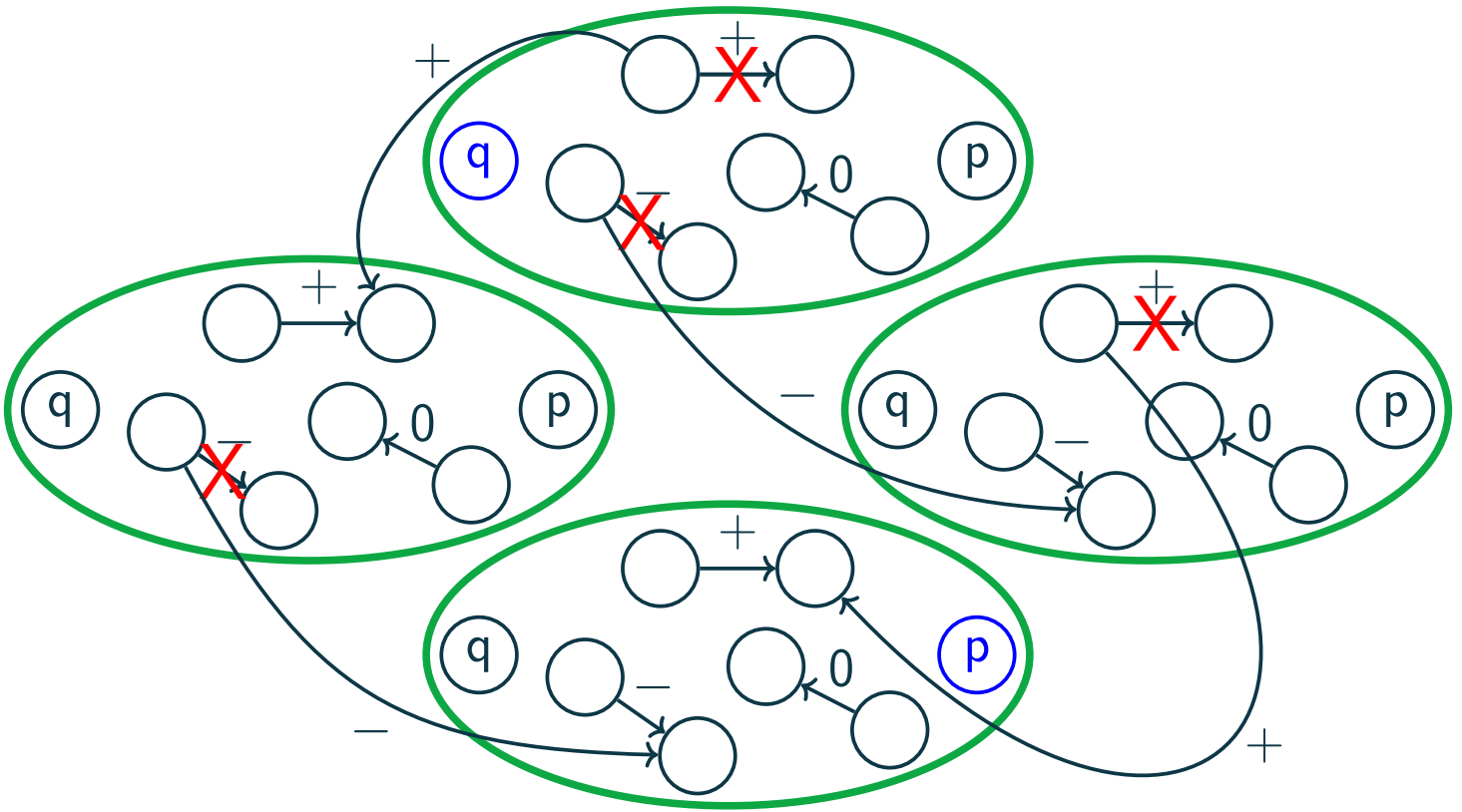
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

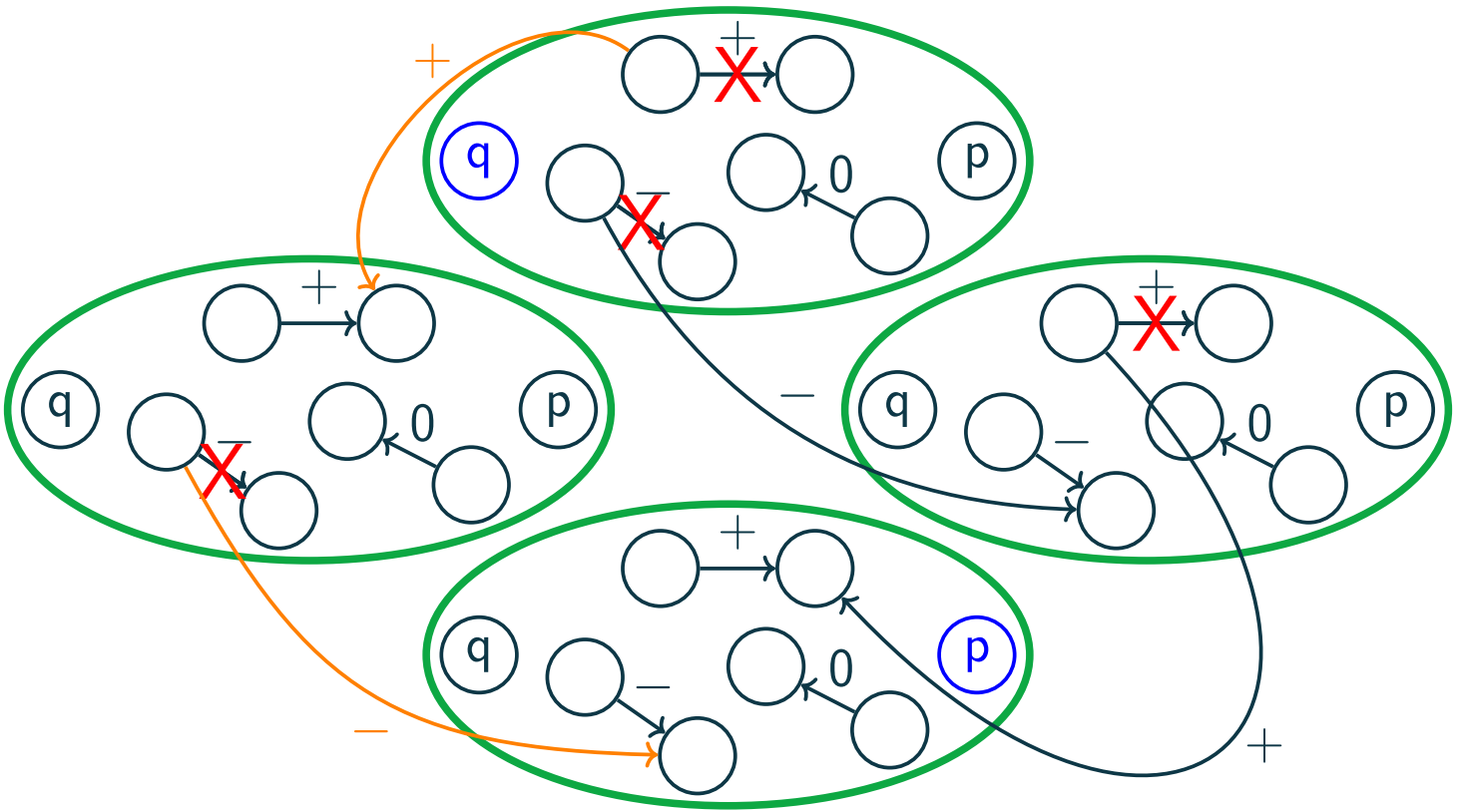
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

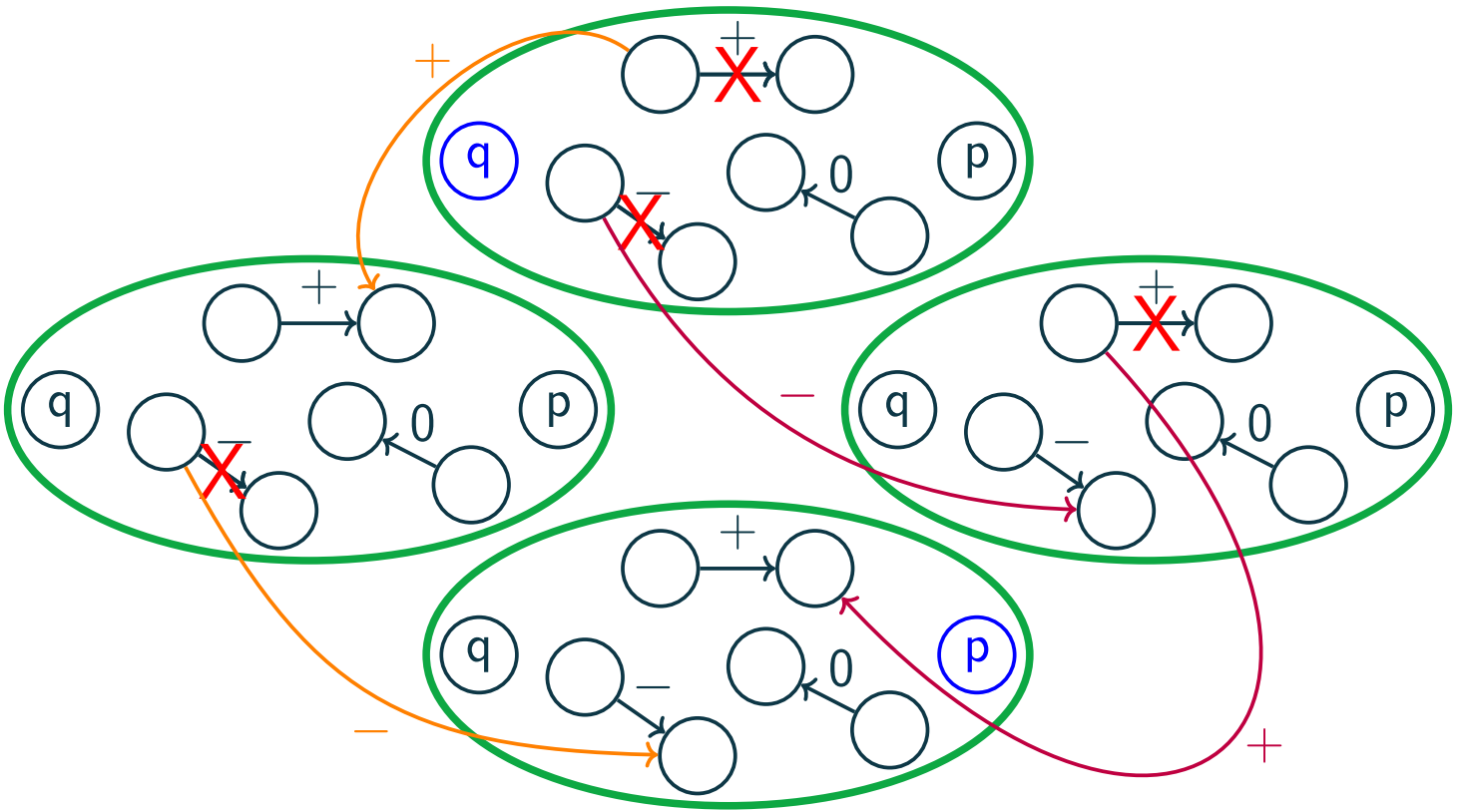
$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

$\Rightarrow$ Checking whether a path from q to p
has positive and negative edges via graph reachability

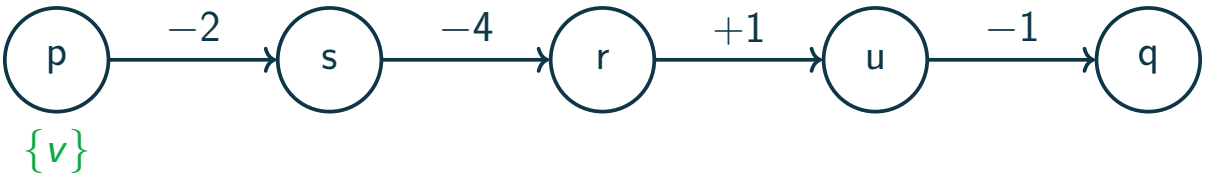


Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

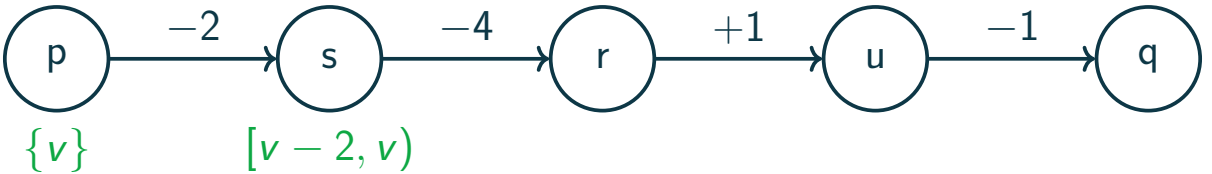
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



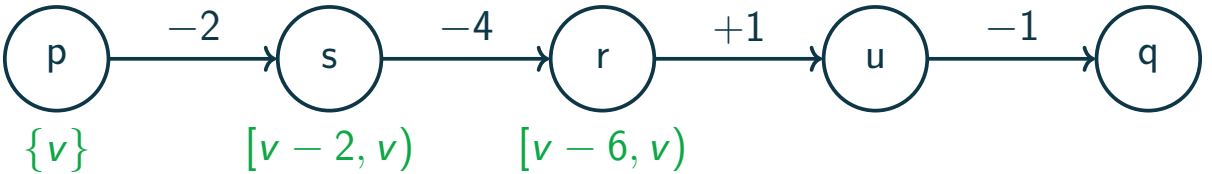
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



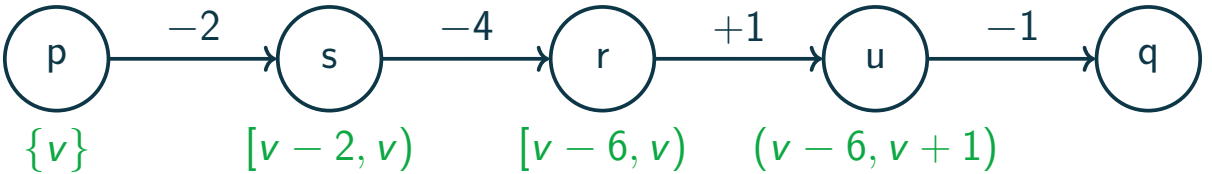
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



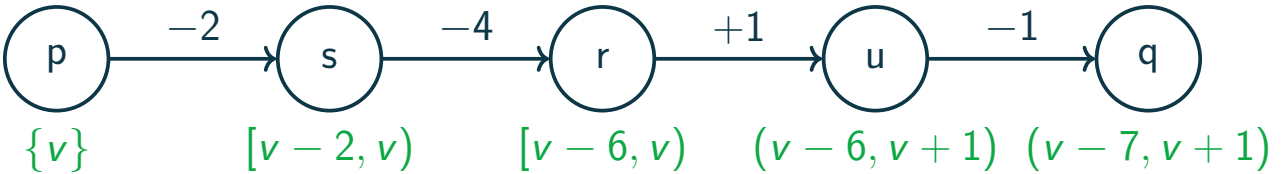
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



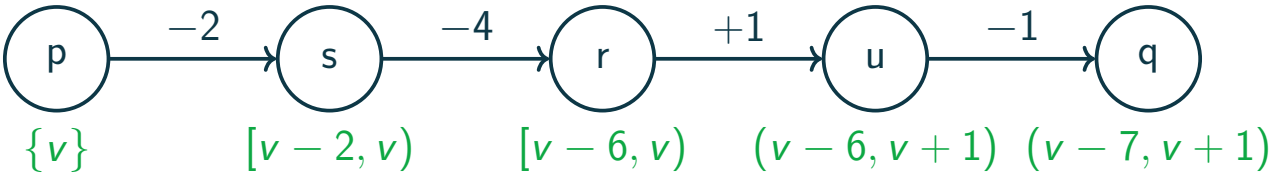
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u

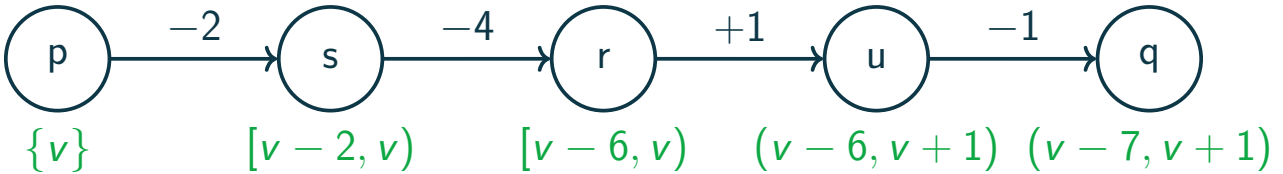


ℓ included if $q(\ell)$ **reachable** by path with **no pos.** edges

u included if $q(u)$ **reachable** by path with **no neg.** edges

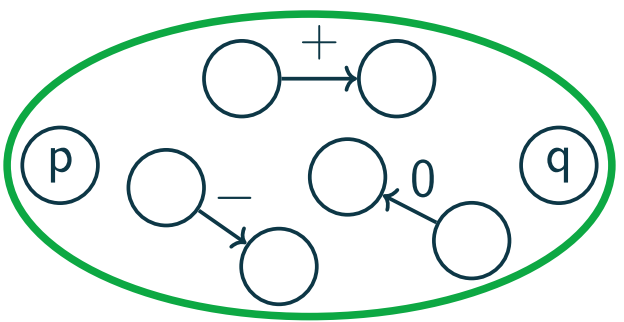
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



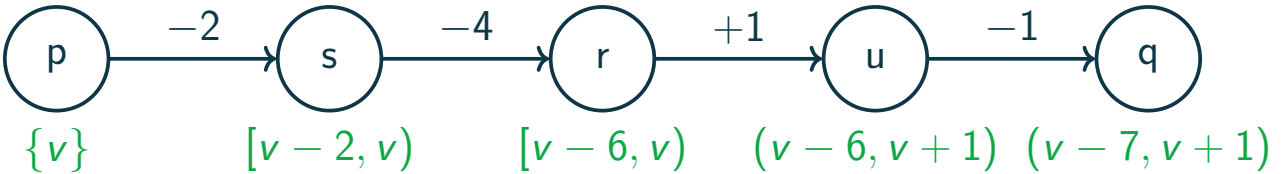
ℓ included if $q(\ell)$ **reachable** by path with **no pos.** edges

u included if $q(u)$ **reachable** by path with **no neg.** edges



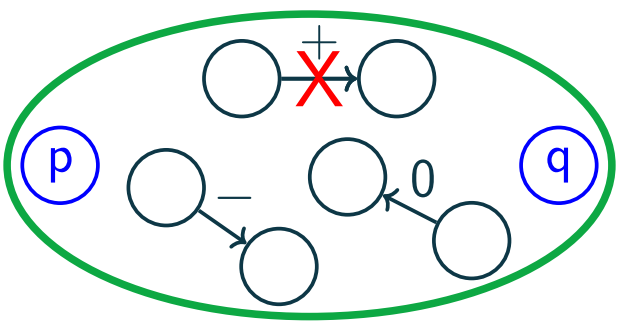
Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



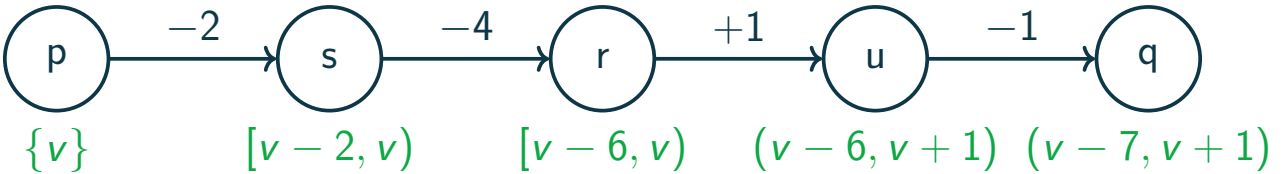
ℓ included if $q(\ell)$ **reachable** by path with **no pos.** edges

u included if $q(u)$ **reachable** by path with **no neg.** edges

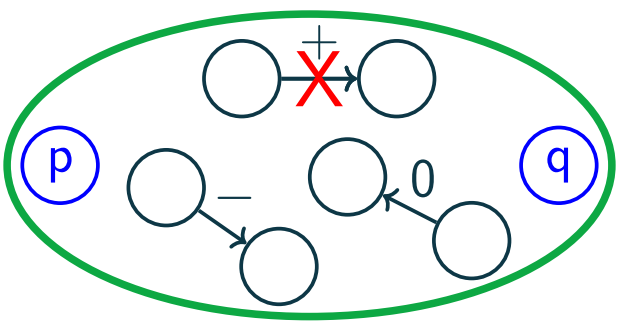


Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



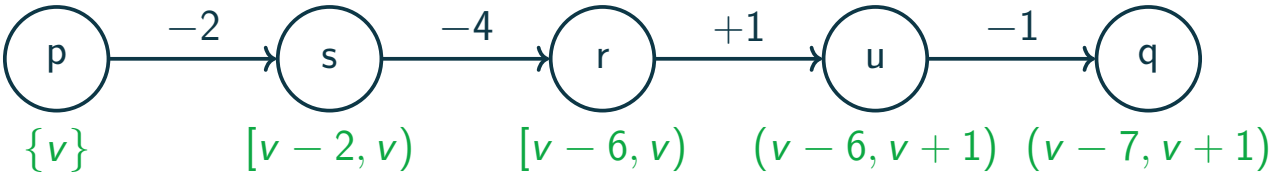
ℓ included if $q(\ell)$ **reachable** by path with **no pos.** edges
u included if $q(u)$ **reachable** by path with **no neg.** edges



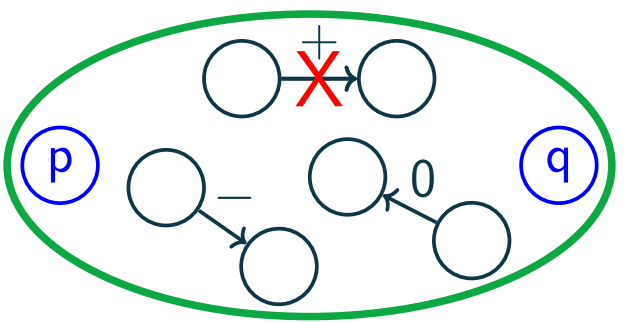
$\Rightarrow$ **Weighted Graph Reachability**!

Guardless COCA

1. Computing ℓ and u 2. Checking membership of ℓ, v, u



ℓ included if $q(\ell)$ **reachable** by path with **no pos.** edges
u included if $q(u)$ **reachable** by path with **no neg.** edges



$\Rightarrow$ **Weighted Graph Reachability**!

$\Rightarrow$ **in NC**^{2}!

Reachability in Continuous OCA (COCA)

COCA have much lower complexity

Guardless OCA:
NP-complete
[Haase et al.,'09]

$\Rightarrow$

Guardless COCA:
in **NC^2** (below P-time)
Even with global guards and equality tests

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

$\Rightarrow$

COCA:
in **P-time**

Parametric OCA:
Decidability unknown

$\Rightarrow$

Parametric COCA:
NP-complete

Reachability in Continuous OCA (COCA)

COCA have much lower complexity

Guardless OCA:
NP-complete
[Haase et al.,'09]

$\Rightarrow$

Guardless COCA:
in **NC^2** (below P-time)
Even with global guards and equality tests

OCA:
PSPACE-complete
[Fearnley,Jurdziński,'15]

$\Rightarrow$

COCA:
in **P-time**

Parametric OCA:
Decidability unknown

$\Rightarrow$

Parametric COCA:
NP-complete

More in our paper: Continuous One-Counter Automata, LICS '21

Summary

Overapproximations help when reachability is **intractable**

Summary

Overapproximations help when reachability is **intractable**

This work:

Overapproximate **One-Counter Automata** via **Continuous One-Counter Automata**

Summary

Overapproximations help when reachability is **intractable**

This work:

Overapproximate **One-Counter Automata** via **Continuous One-Counter Automata**

Reachability sets of Continuous One-Counter Automata are **unions** of **few** intervals

$\Rightarrow$ **tractable reachability**

Summary

Overapproximations help when reachability is **intractable**

This work:

Overapproximate **One-Counter Automata** via **Continuous One-Counter Automata**

Reachability sets of Continuous One-Counter Automata are **unions** of **few** intervals

$\Rightarrow$ **tractable reachability**

Thanks for listening!