The Reachability Problem for Vector Addition Systems

Wojciech Czerwiński

Aalborg 2022

Plan

Plan

- basic notions and the problem

Plan

- basic notions and the problem
- short history

Plan

- basic notions and the problem
- short history
- interesting examples

Plan

- basic notions and the problem
- short history
- interesting examples
- decidability (idea)

Plan

- basic notions and the problem
- short history
- interesting examples
- decidability (idea)
- Ackermann-hardness

Plan

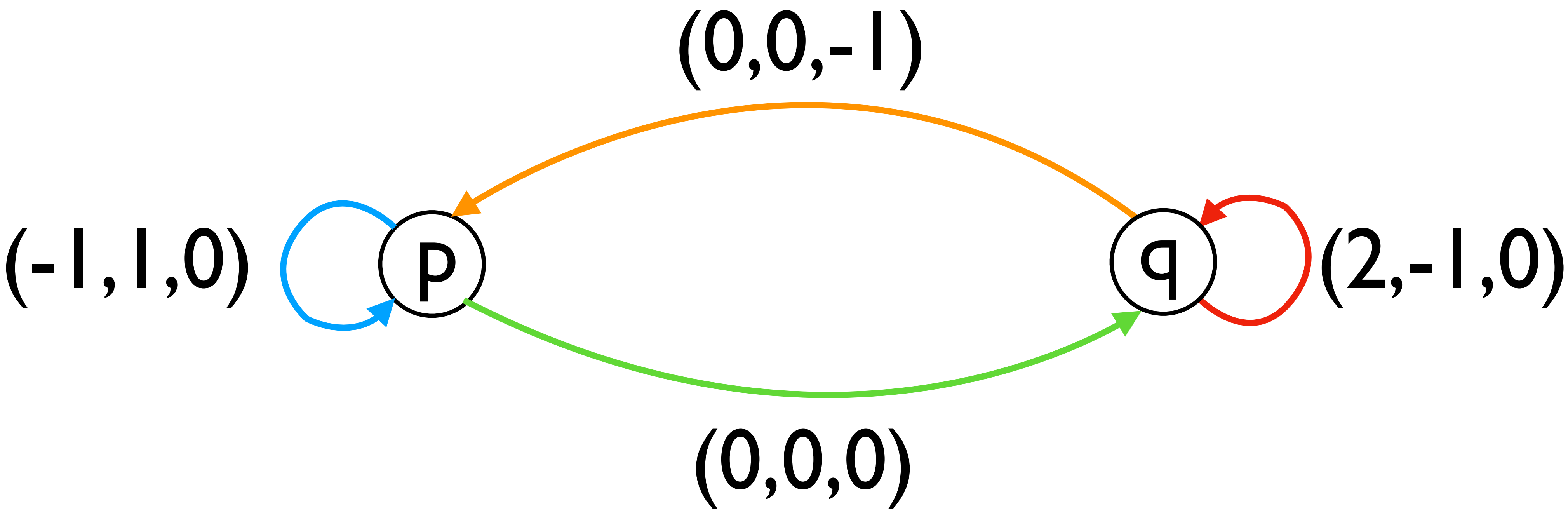
- basic notions and the problem
- short history
- interesting examples
- decidability (idea)
- Ackermann-hardness
- open problems

Plan

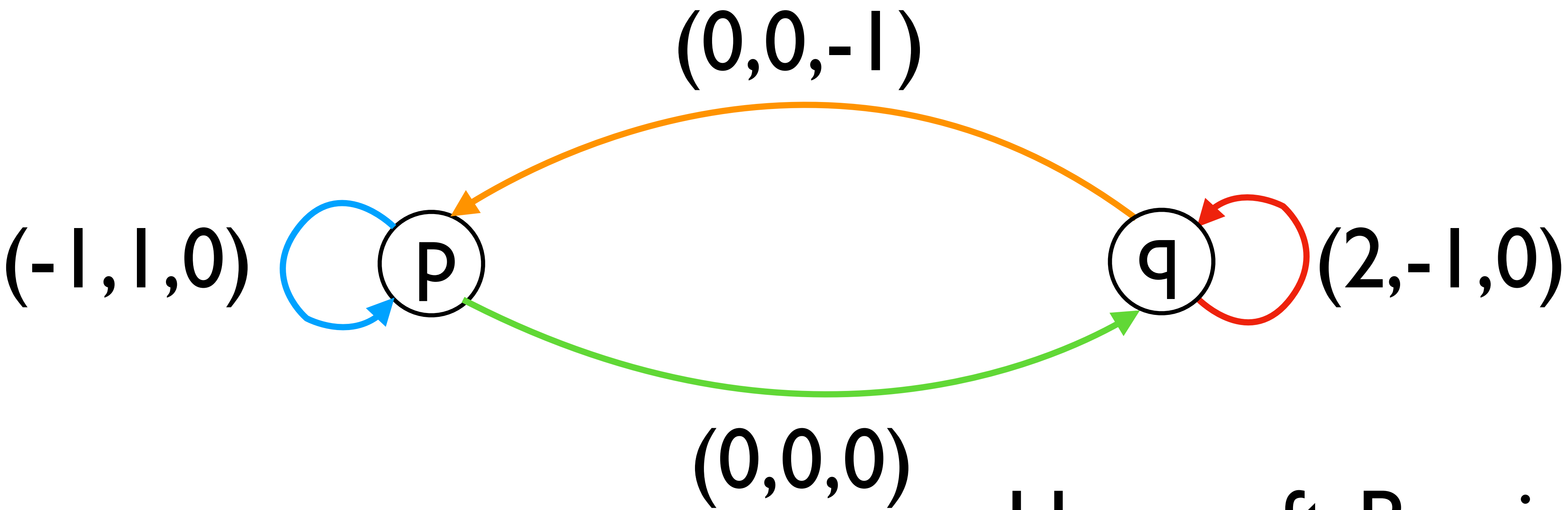
- basic notions and the problem
- short history
- interesting examples
- decidability (idea)
- Ackermann-hardness
- open problems
- goal: intuitions

Vector Addition Systems with States

Vector Addition Systems with States

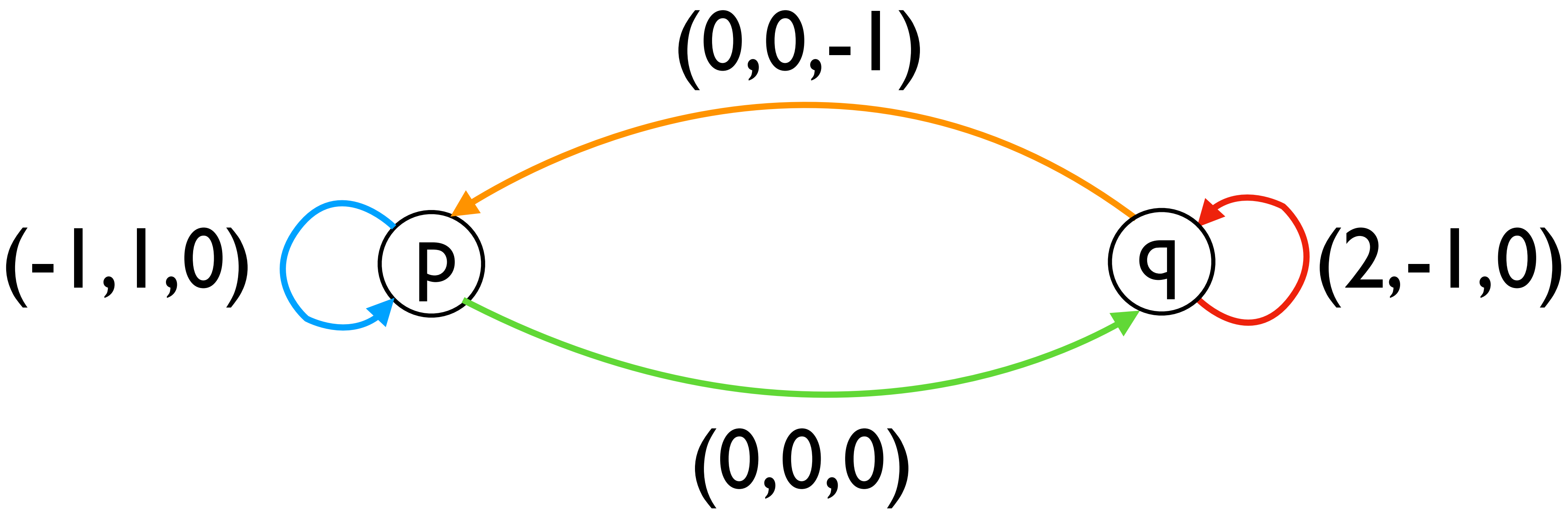


Vector Addition Systems with States

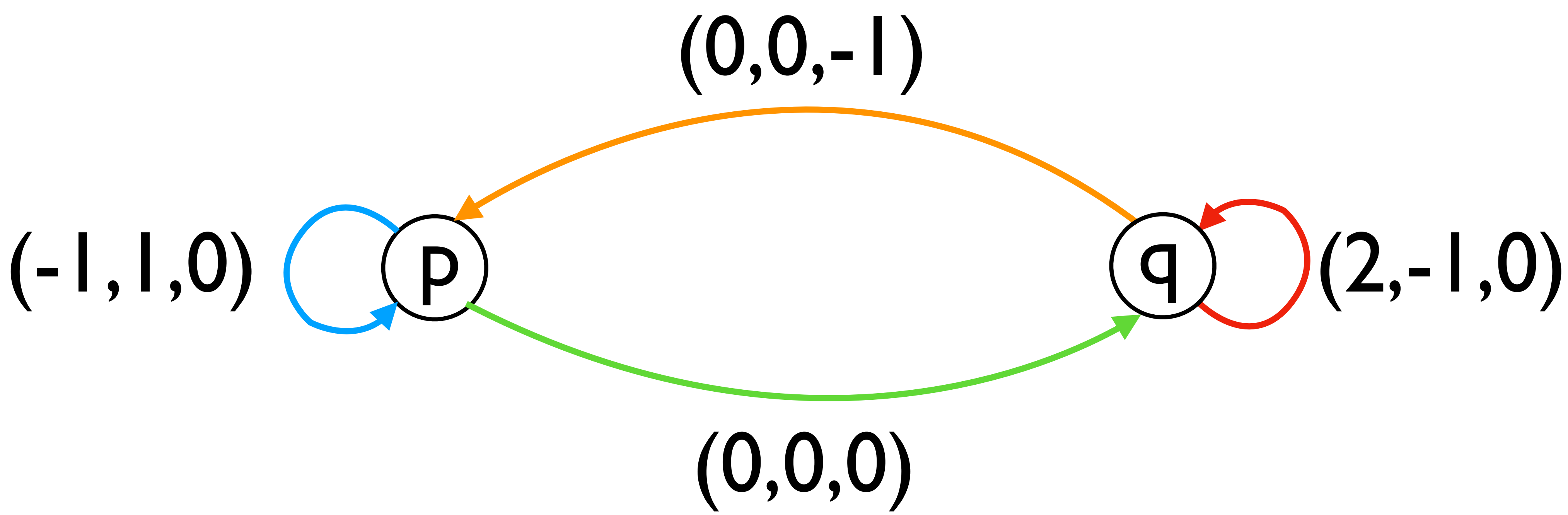


Hopcroft-Pansiot `78

Vector Addition Systems with States

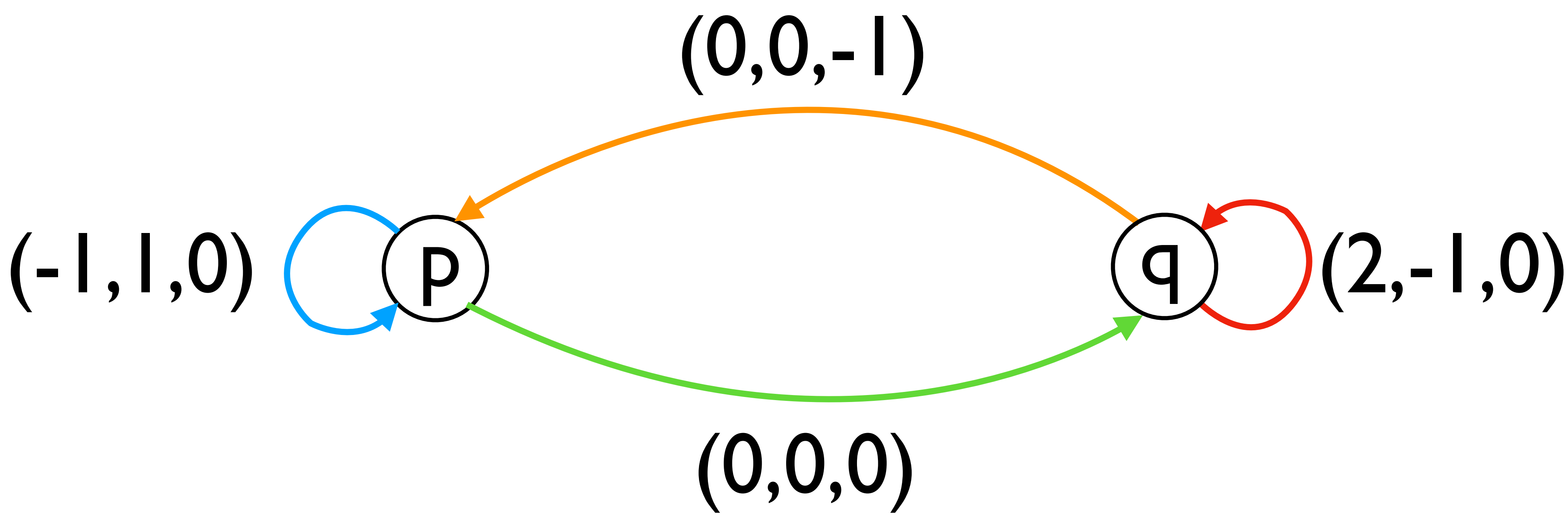


Vector Addition Systems with States



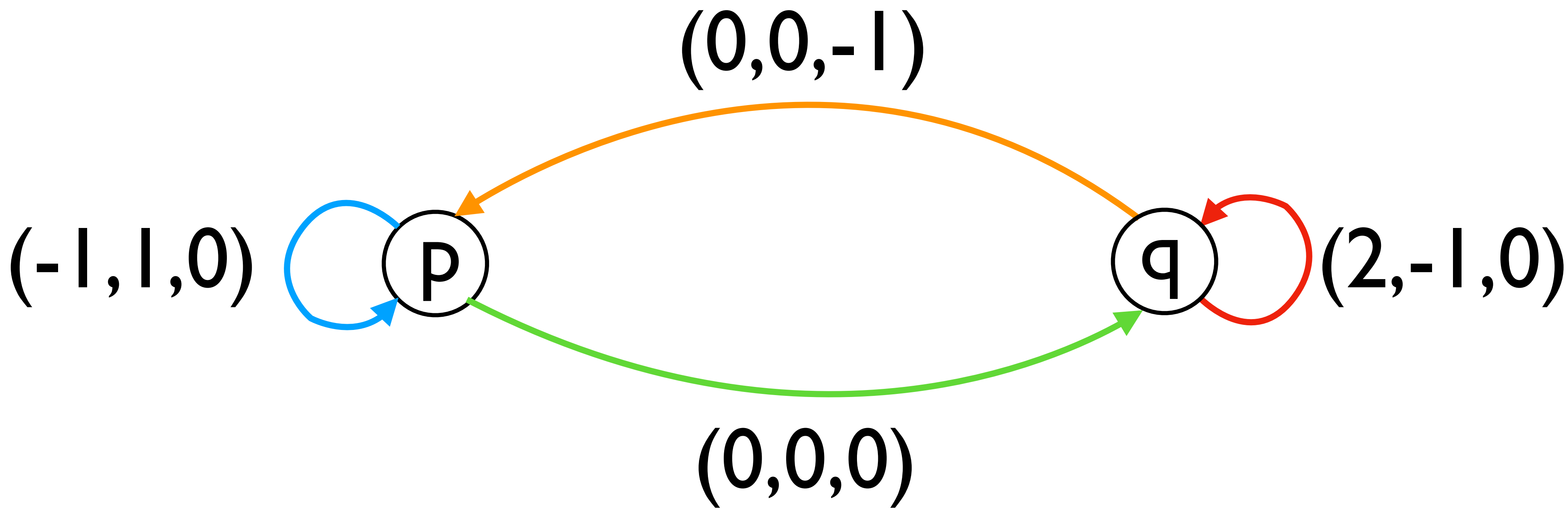
p(2,0,7)

Vector Addition Systems with States



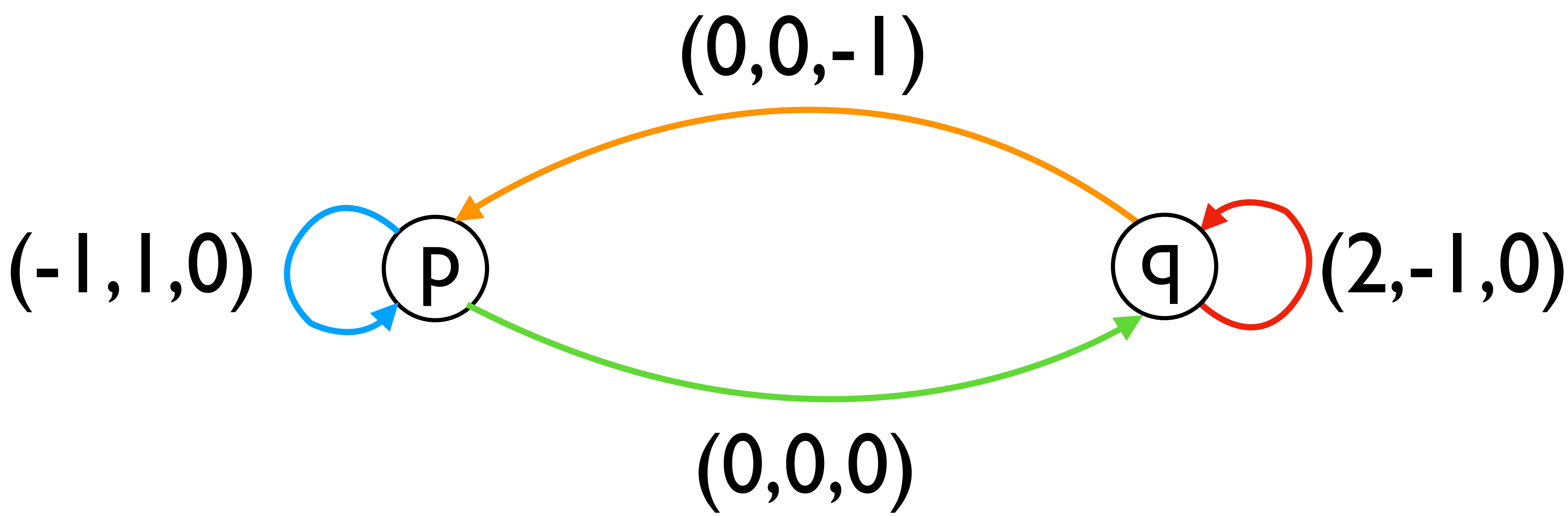
p(2,0,7) → p(1,1,7)

Vector Addition Systems with States



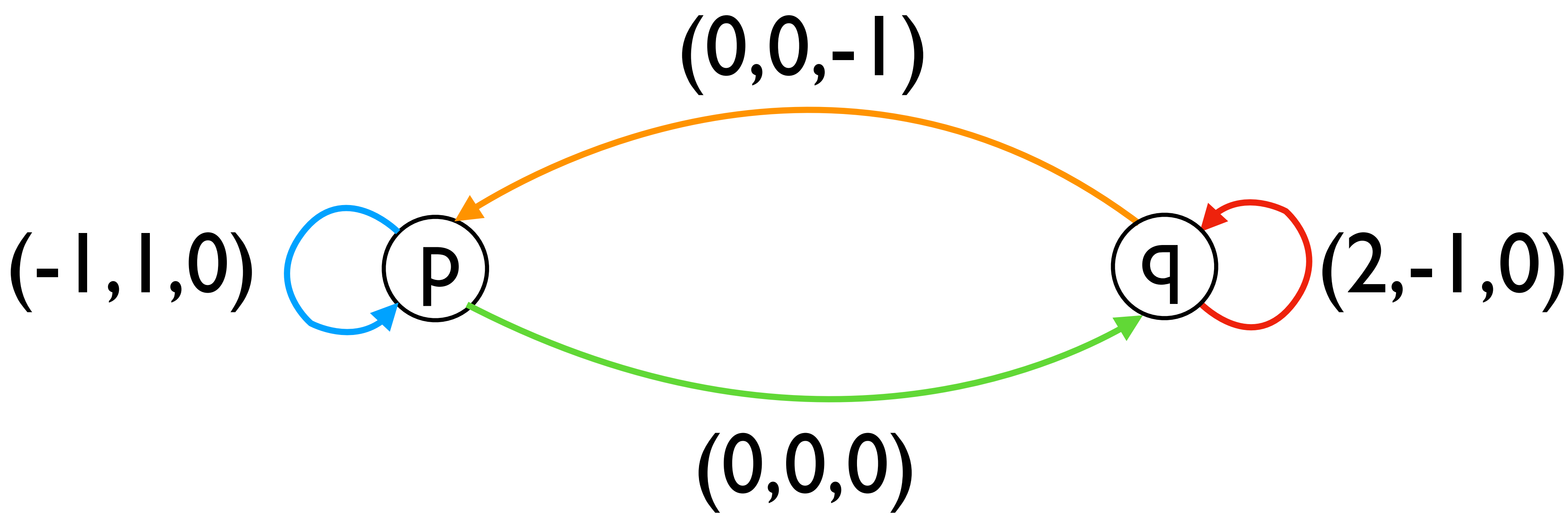
p(2,0,7) → p(1,1,7) → p(0,2,7)

Vector Addition Systems with States



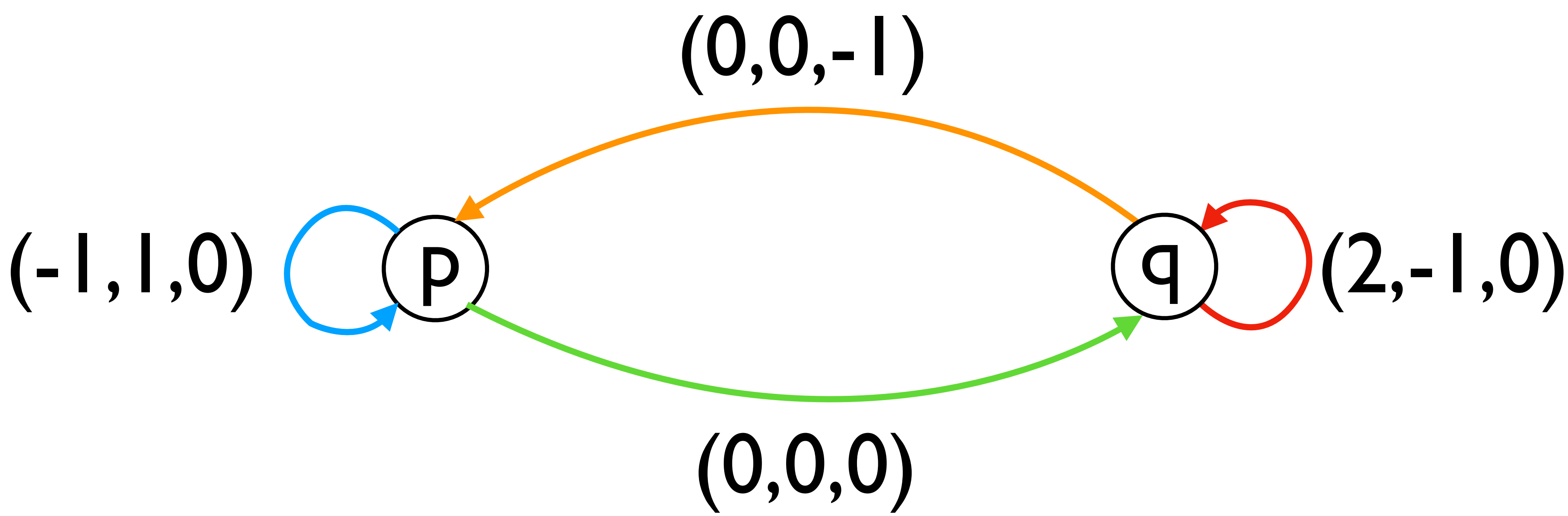
p(2,0,7) → p(1,1,7) → p(0,2,7) → q(0,2,7)

Vector Addition Systems with States



p(2,0,7) → p(1,1,7) → p(0,2,7) → q(0,2,7) → q(2,1,7)

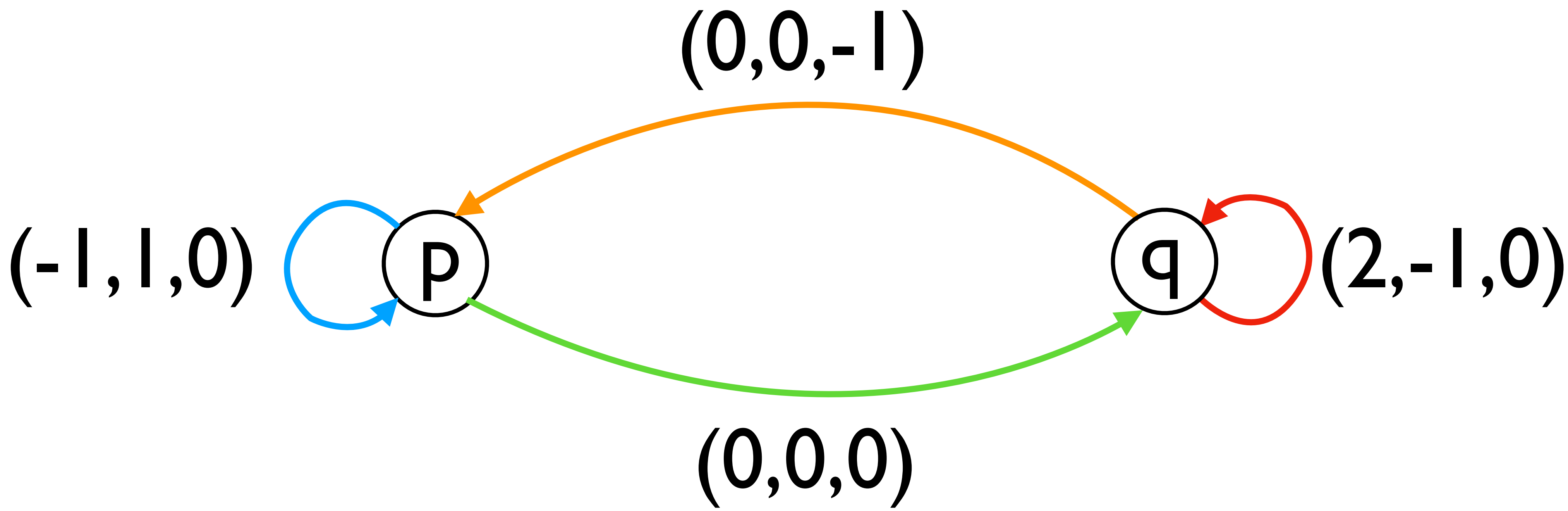
Vector Addition Systems with States



p(2,0,7) → p(1,1,7) → p(0,2,7) → q(0,2,7) → q(2,1,7)

→ q(4,0,7)

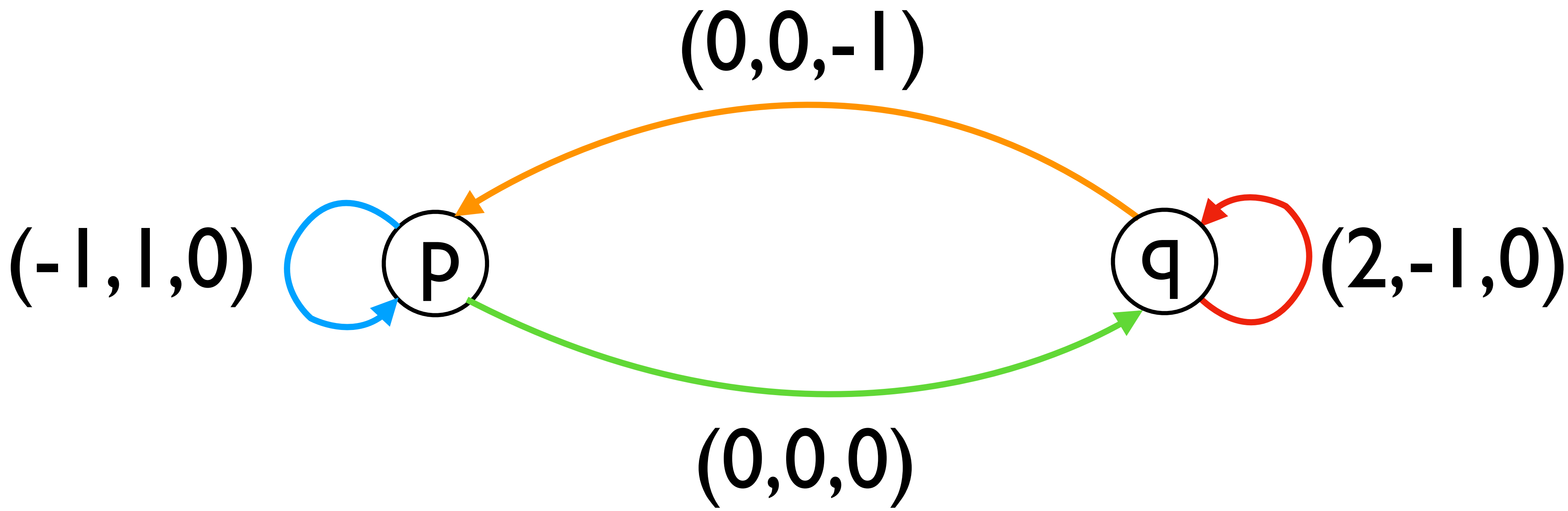
Vector Addition Systems with States



p(2,0,7) → p(1,1,7) → p(0,2,7) → q(0,2,7) → q(2,1,7)

→ q(4,0,7) → p(4,0,6)

Vector Addition Systems with States

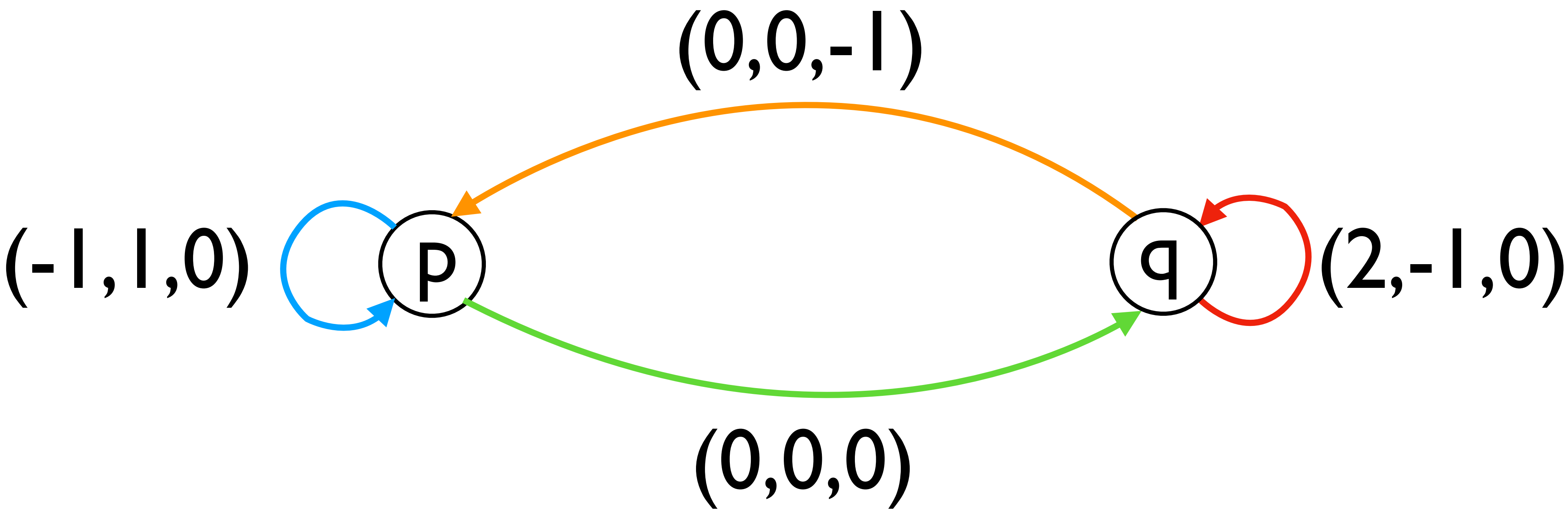


p(2,0,7) → p(1,1,7) → p(0,2,7) → q(0,2,7) → q(2,1,7)

→ q(4,0,7) → p(4,0,6)

Petri nets

Vector Addition Systems with States

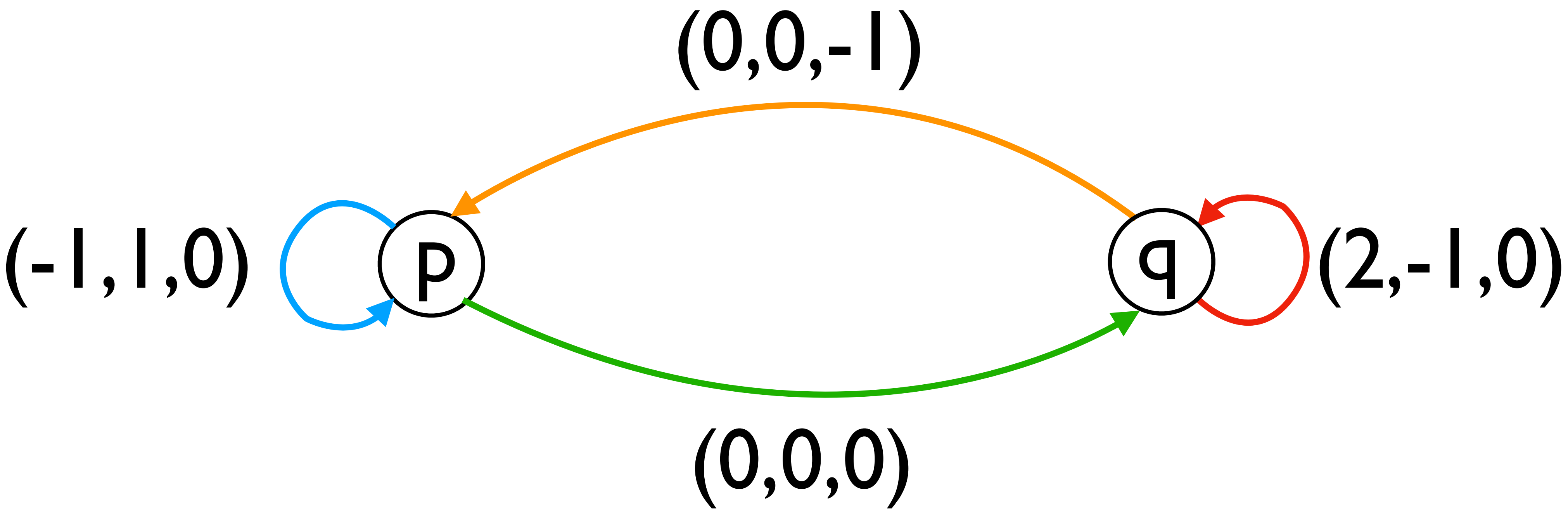


p(2,0,7) → p(1,1,7) → p(0,2,7) → q(0,2,7) → q(2,1,7)

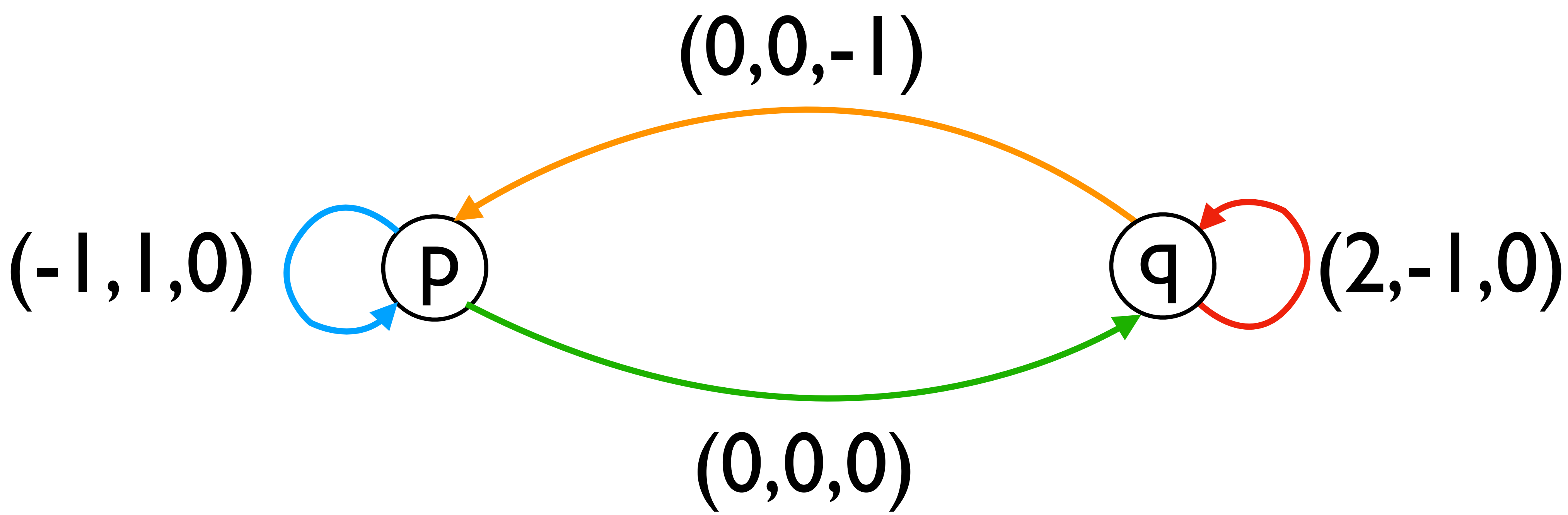
→ q(4,0,7) → p(4,0,6)

Vector Addition Systems with States

Vector Addition Systems with States

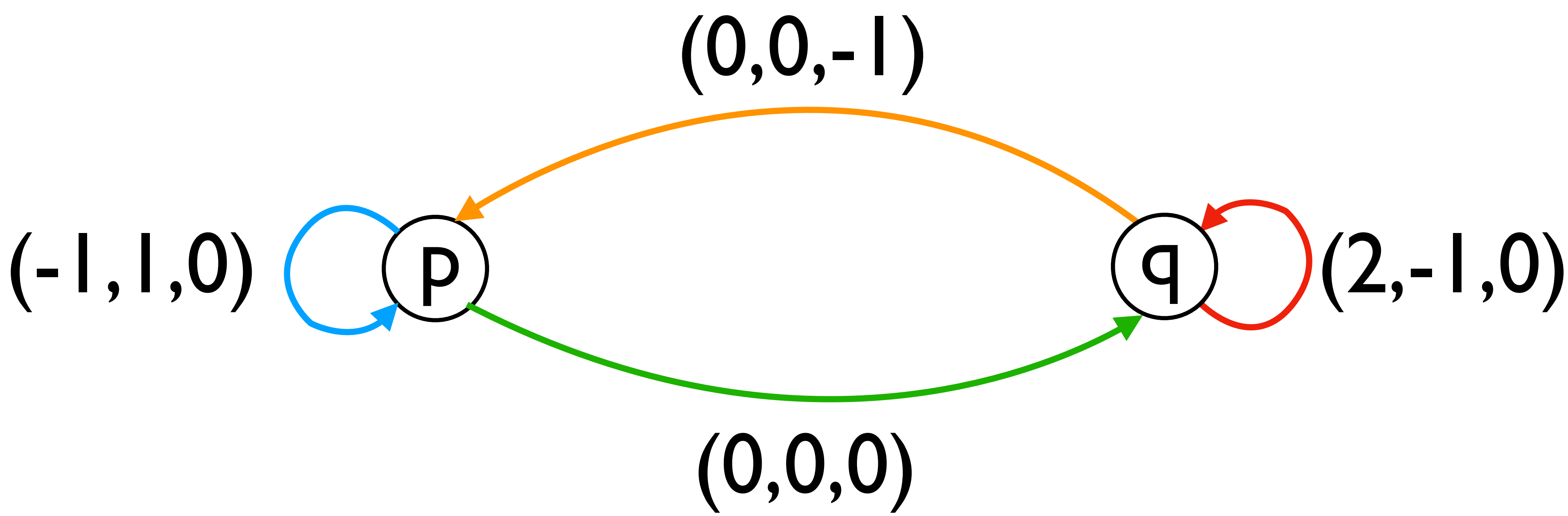


Vector Addition Systems with States



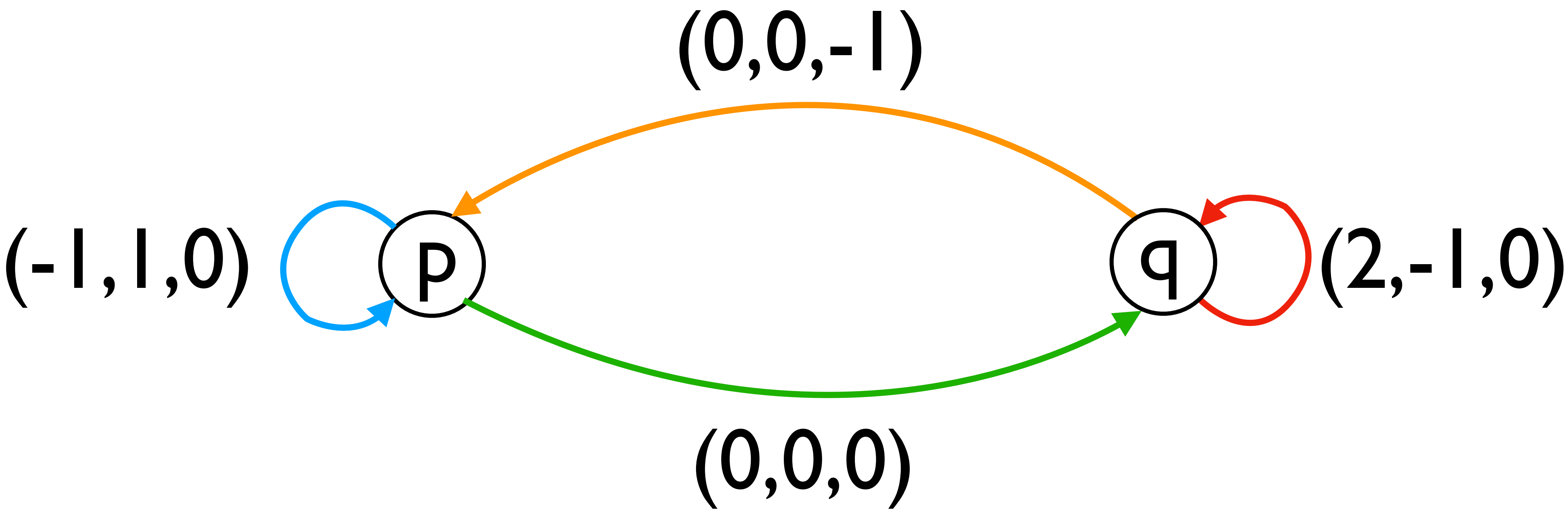
p(k,0,n)

Vector Addition Systems with States



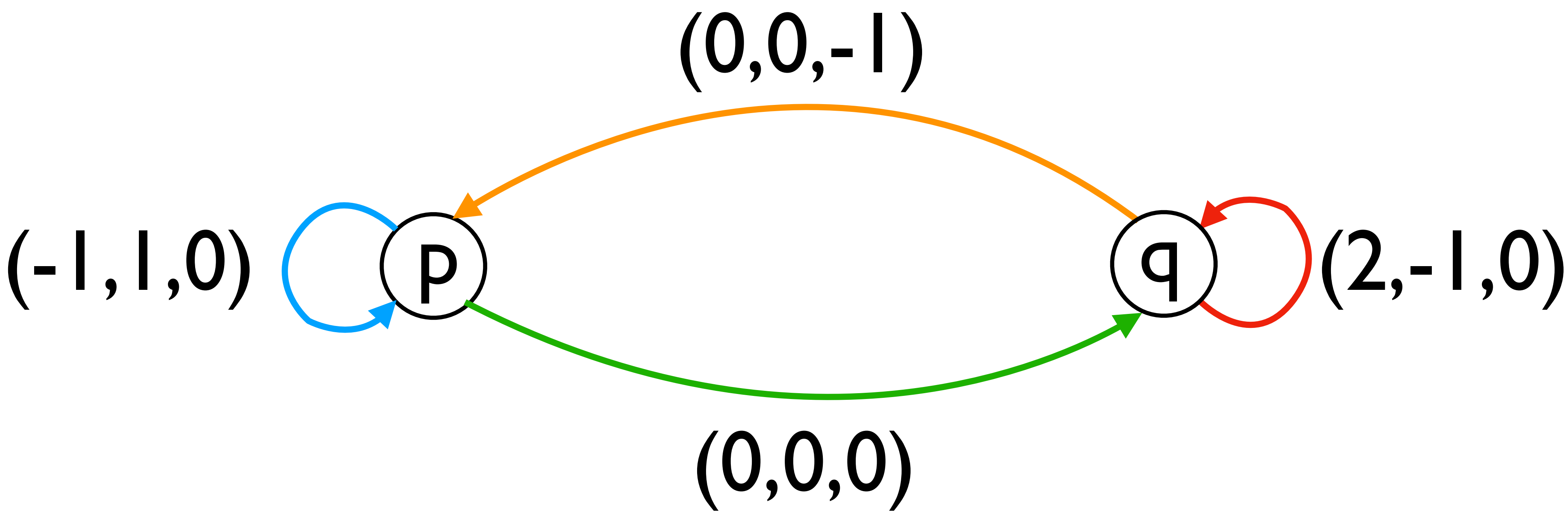
p(k,0,n) ⟶ p(0,k,n)

Vector Addition Systems with States



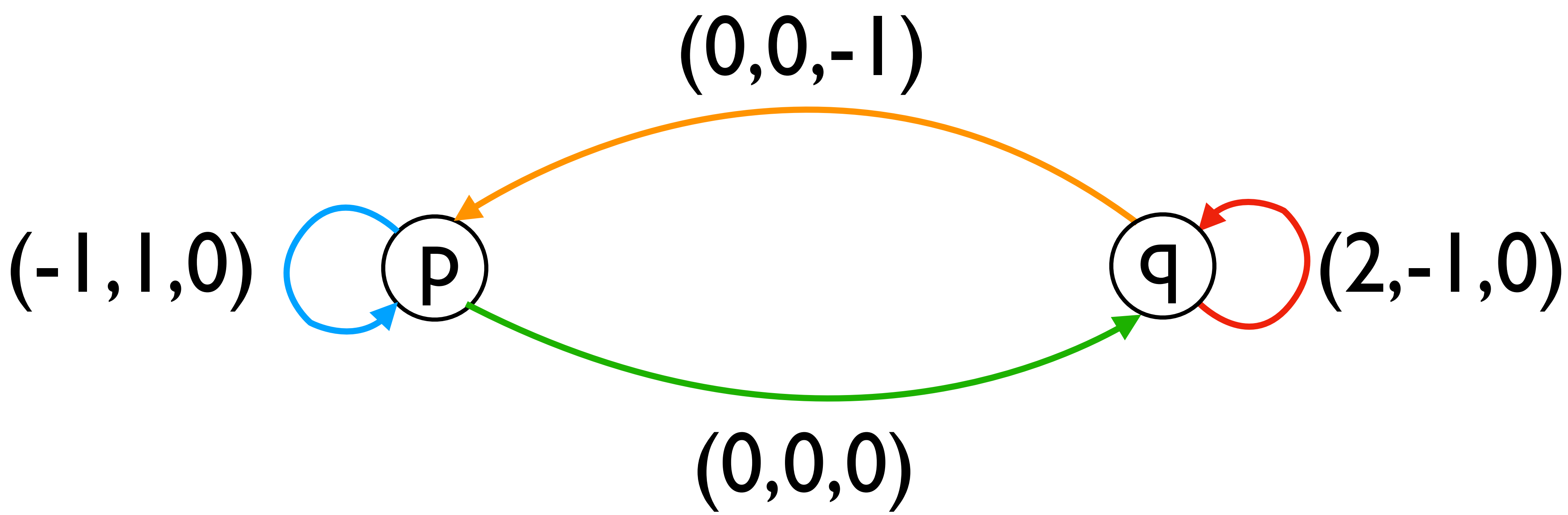
$p(k,0,n) \longrightarrow p(0,k,n) \longrightarrow q(0,k,n)$

Vector Addition Systems with States



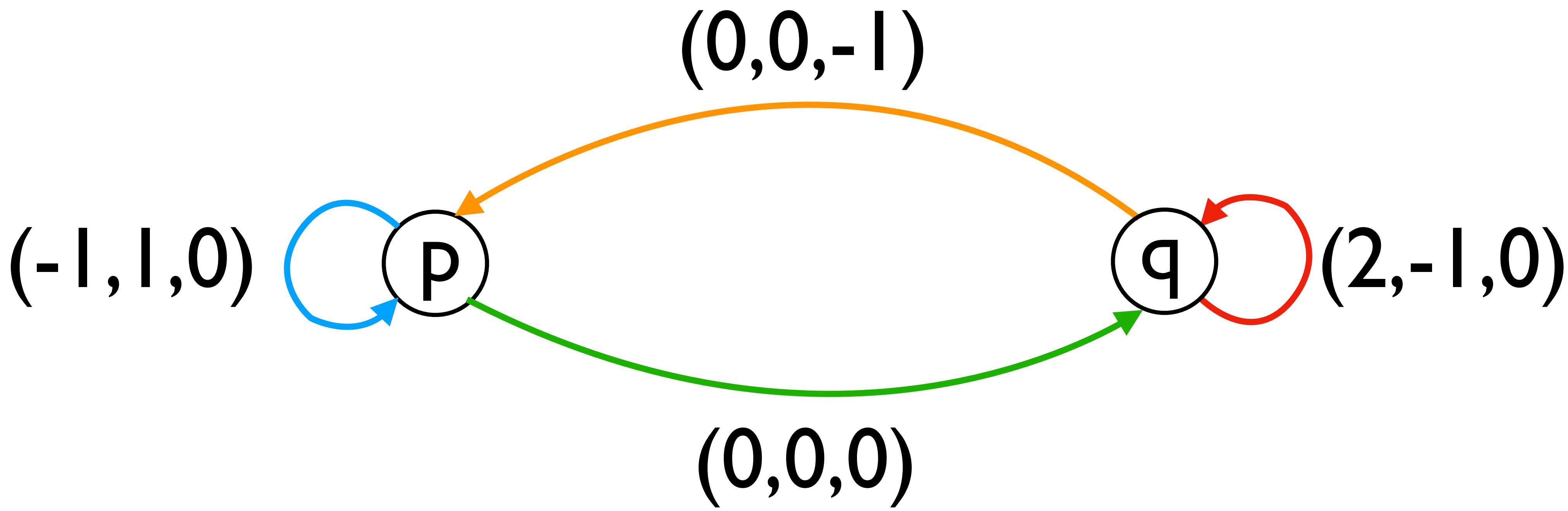
p(k,0,n) → p(0,k,n) → q(0,k,n) → q(2k,0,n)

Vector Addition Systems with States



p(k,0,n) → p(0,k,n) → q(0,k,n) → q(2k,0,n) → p(2k,0,n-1)

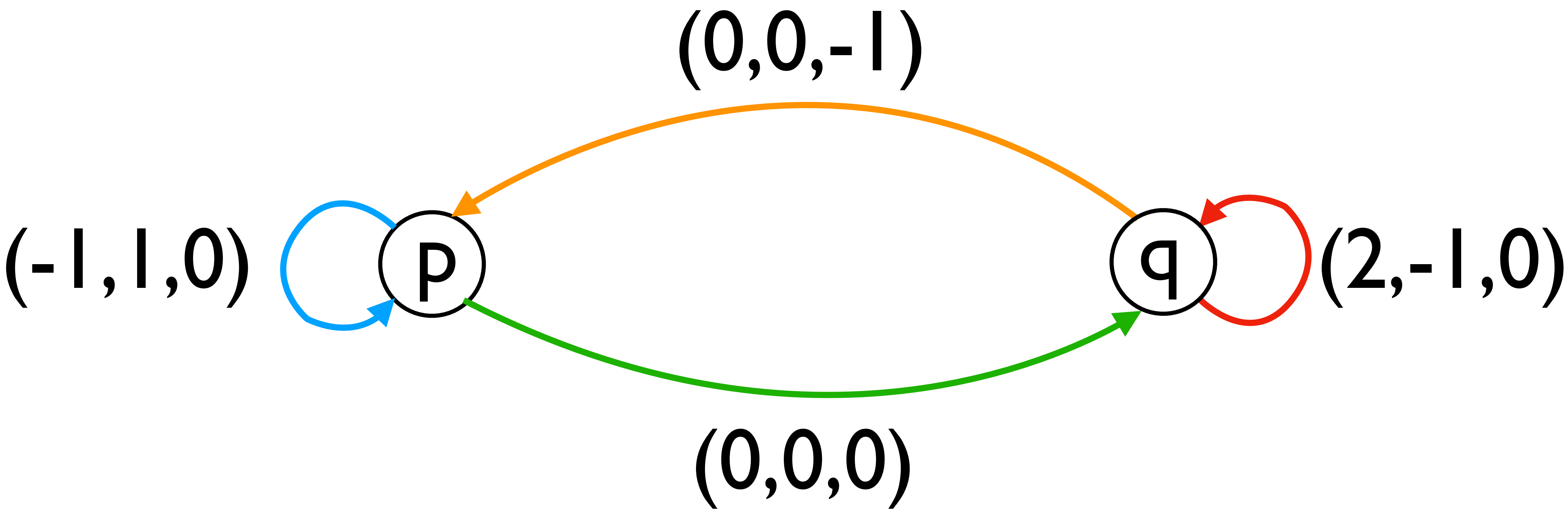
Vector Addition Systems with States



p(k,0,n) → p(0,k,n) → q(0,k,n) → q(2k,0,n) → p(2k,0,n-1)

p(1,0,n)

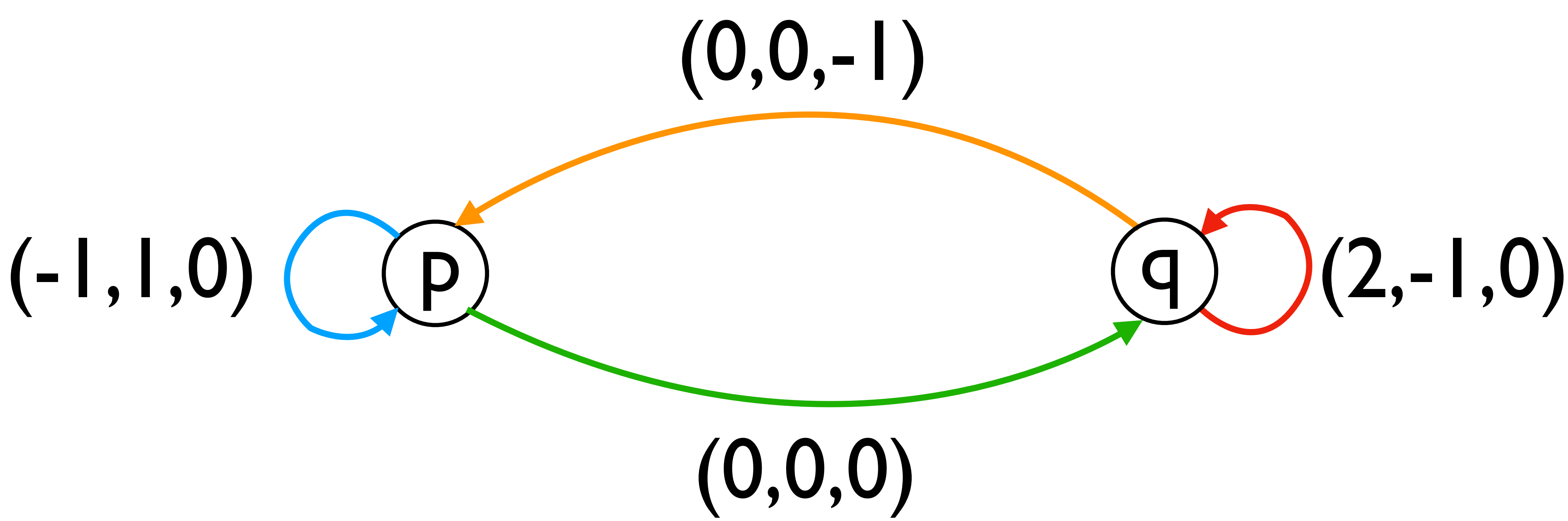
Vector Addition Systems with States



$p(k,0,n) \rightarrow p(0,k,n) \rightarrow q(0,k,n) \rightarrow q(2k,0,n) \rightarrow p(2k,0,n-1)$

$p(1,0,n) \longrightarrow p(2,0,n-1)$

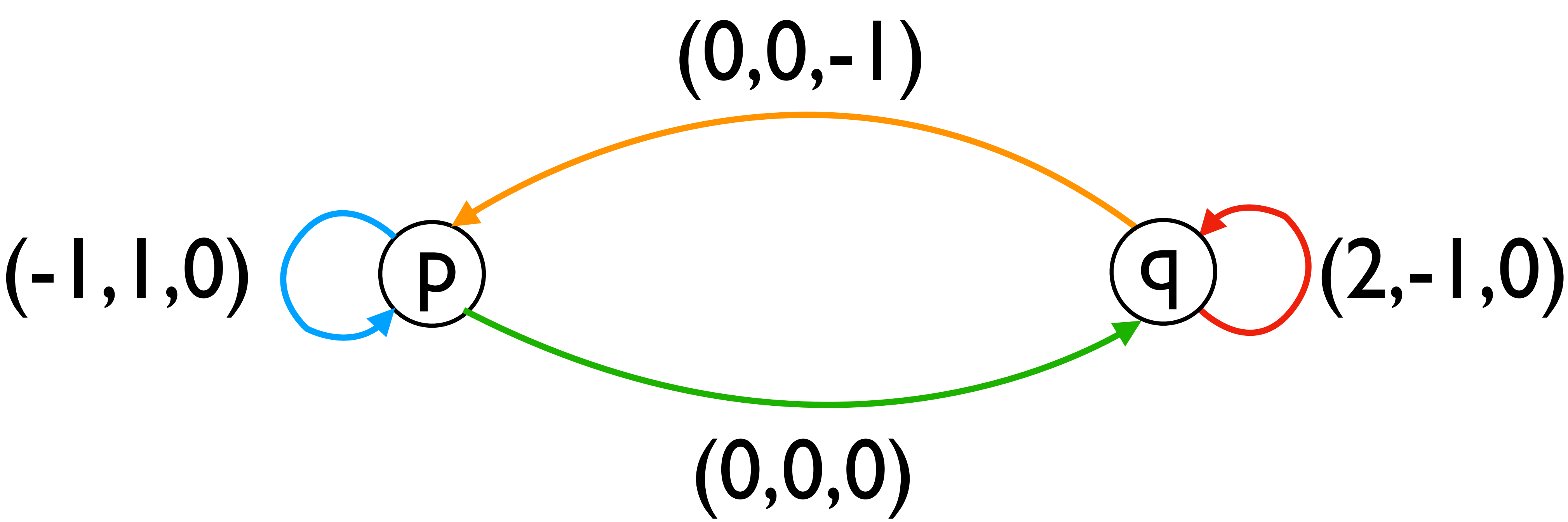
Vector Addition Systems with States



p(k,0,n) → p(0,k,n) → q(0,k,n) → q(2k,0,n) → p(2k,0,n-1)

p(1,0,n) → p(2,0,n-1) …

Vector Addition Systems with States



$$p(k,0,n) \rightarrow p(0,k,n) \rightarrow q(0,k,n) \rightarrow q(2k,0,n) \rightarrow p(2k,0,n-1)$$

$$p(1,0,n) \rightarrow p(2,0,n-1) \;\ldots\; \rightarrow p(2^n,0,0)$$

Reachability problem

Reachability problem

Given: a VASS, two its configurations s and t

Reachability problem

Given: a VASS, two its configurations s and t

Question: is there a run from s to t?

Reachability problem

Given: a VASS, two its configurations s and t

Question: is there a run from s to t?

Question: is there a run from s above t?

Reachability problem

Given: a VASS, two its configurations s and t

Question: is there a run from s to t?

Question: is there a run from s above t?

coverability problem

Short history

Short history

Lipton `76: ExpSpace-hardness of coverability

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

doubly-exponential length paths

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

Mayr `81: decidability of reachability

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

Mayr `81: decidability of reachability

Kosaraju `82, Lambert `92: simplifications

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

Mayr `81: decidability of reachability

Kosaraju `82, Lambert `92: simplifications

Blondin at el. `15: reachability PSpace-complete for 2-VASSes

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

Mayr `81: decidability of reachability

Kosaraju `82, Lambert `92: simplifications

Blondin at el. `15: reachability PSpace-complete
for 2-VASSes

exponential length paths

Short history

Lipton `76: ExpSpace-hardness of coverability

Rackoff `78: coverability in ExpSpace

Mayr `81: decidability of reachability

Kosaraju `82, Lambert `92: simplifications

Blondin at el. `15: reachability PSpace-complete for 2-VASSes

Short history

Short history

Leroux, Schmitz `15: cubic-Ackermann

Short history

Leroux, Schmitz `15: cubic-Ackermann

Leroux, Schmitz `19: Ackermann

Short history

Leroux, Schmitz `15: cubic-Ackermann

Leroux, Schmitz `19: Ackermann

Conjecture: reachability in ExpSpace

Short history

Leroux, Schmitz `15: cubic-Ackermann

Leroux, Schmitz `19: Ackermann

Conjecture: reachability in ExpSpace

Cz., Lasota, Lazic, Leroux, Mazowiecki `19:
Tower-hardness

Short history

Leroux, Schmitz `15: cubic-Ackermann

Leroux, Schmitz `19: Ackermann

Conjecture: reachability in ExpSpace

Cz., Lasota, Lazic, Leroux, Mazowiecki `19:
Tower-hardness

Leroux & Cz., Orlikowski`21: Ackermann-hardness

Functions F_k

Functions F_k

$F_1(n) = 2n$

Functions F_k

$F_1(n) = 2n$

$F_{k+1}(n) = F_k \circ \ldots \circ F_k(1)$

Functions F_k

$F_1(n) = 2n$

$F_{k+1}(n) = F_k \circ \ldots \circ F_k(1)$

composed n times

Functions F_k

$F_1(n) = 2n$

$F_{k+1}(n) = F_k \circ \ldots \circ F_k(1)$

composed n times

$F_2(n) = 2^n$

Functions F_k

$F_1(n) = 2n$

$F_{k+1}(n) = F_k \circ \ldots \circ F_k(1)$

composed n times

$F_2(n) = 2^n$

$F_3(n) = \text{Tower}(n)$

Functions F_k

$F_1(n) = 2n$

$F_{k+1}(n) = F_k \circ \ldots \circ F_k(1)$

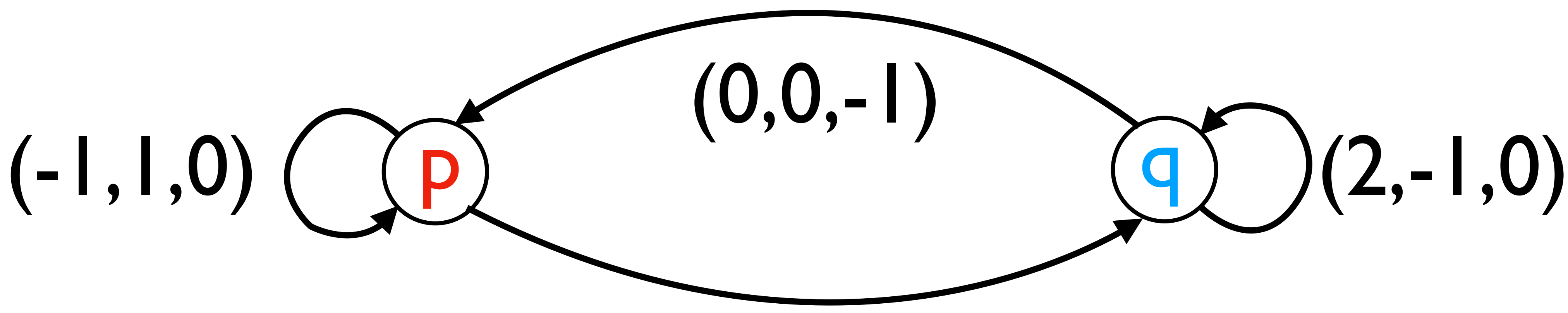
composed n times

$F_2(n) = 2^n$ $F_3(n) = \text{Tower}(n)$

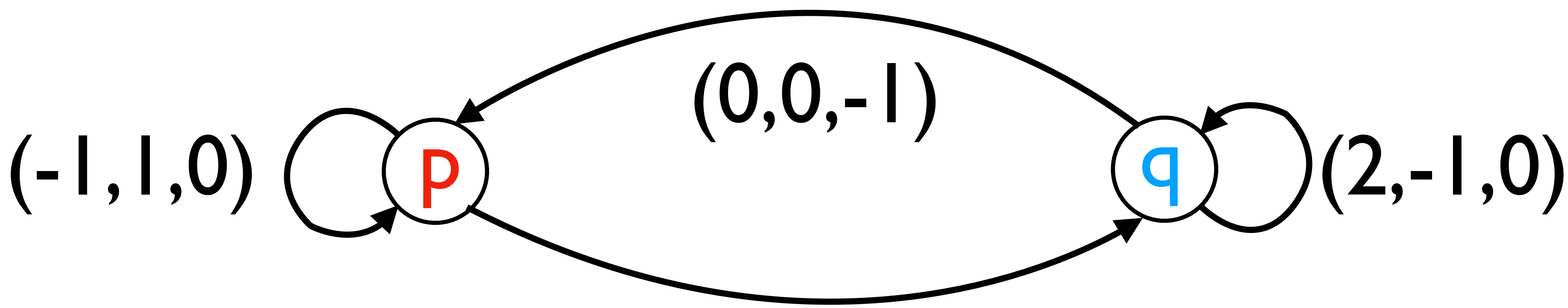
$\text{Ack}(n) = F_\omega(n) = F_n(n)$

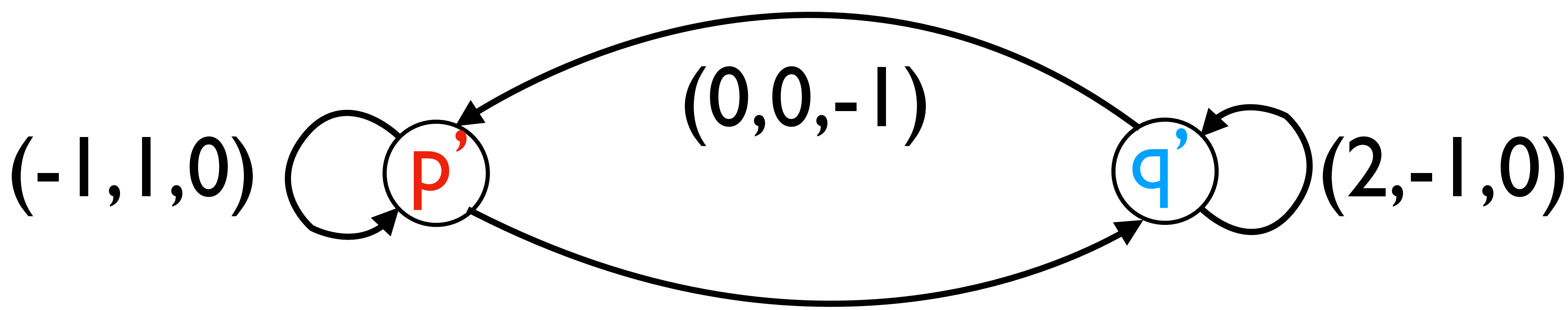
Hard examples

Hard examples

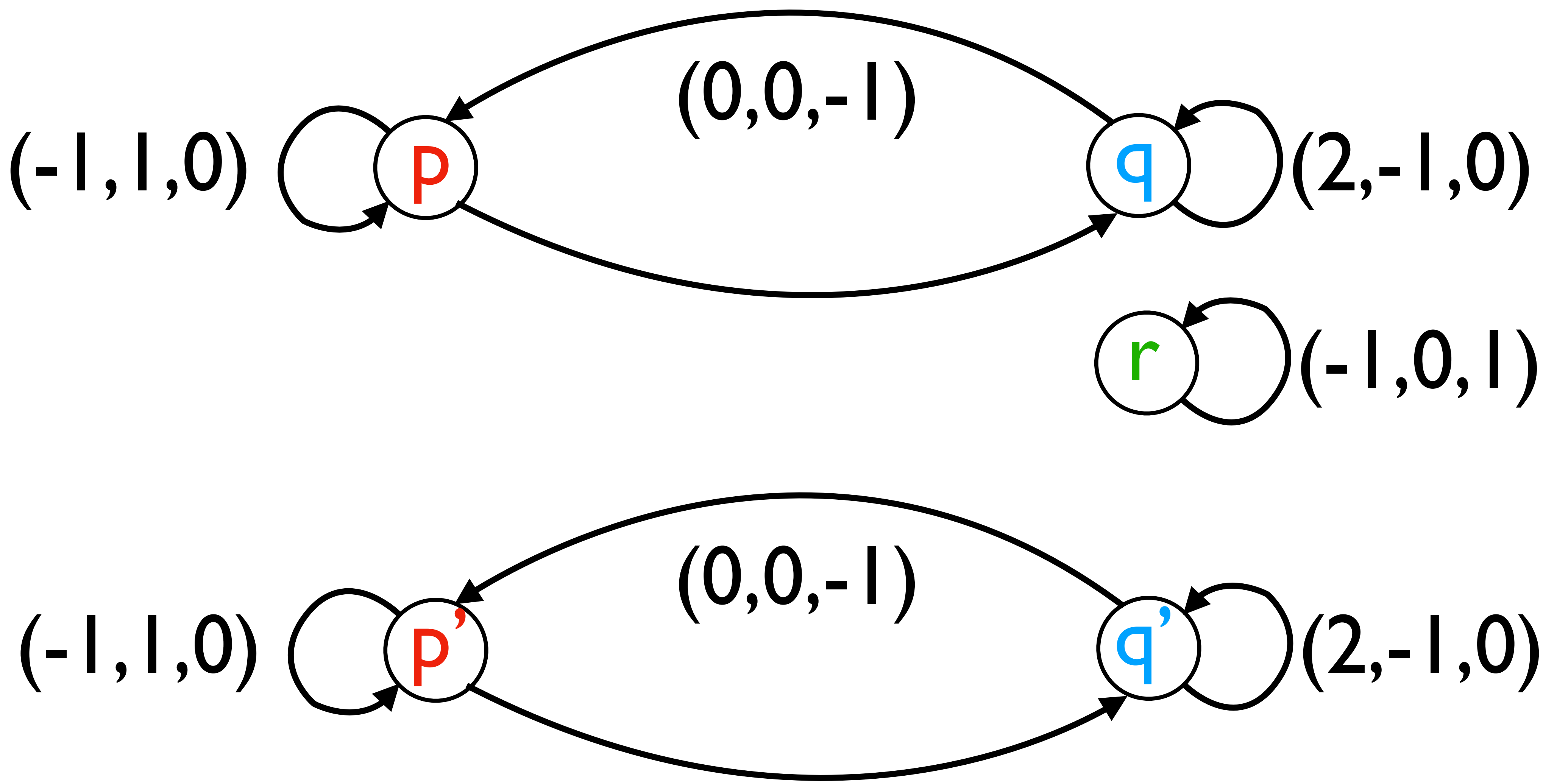


Hard examples

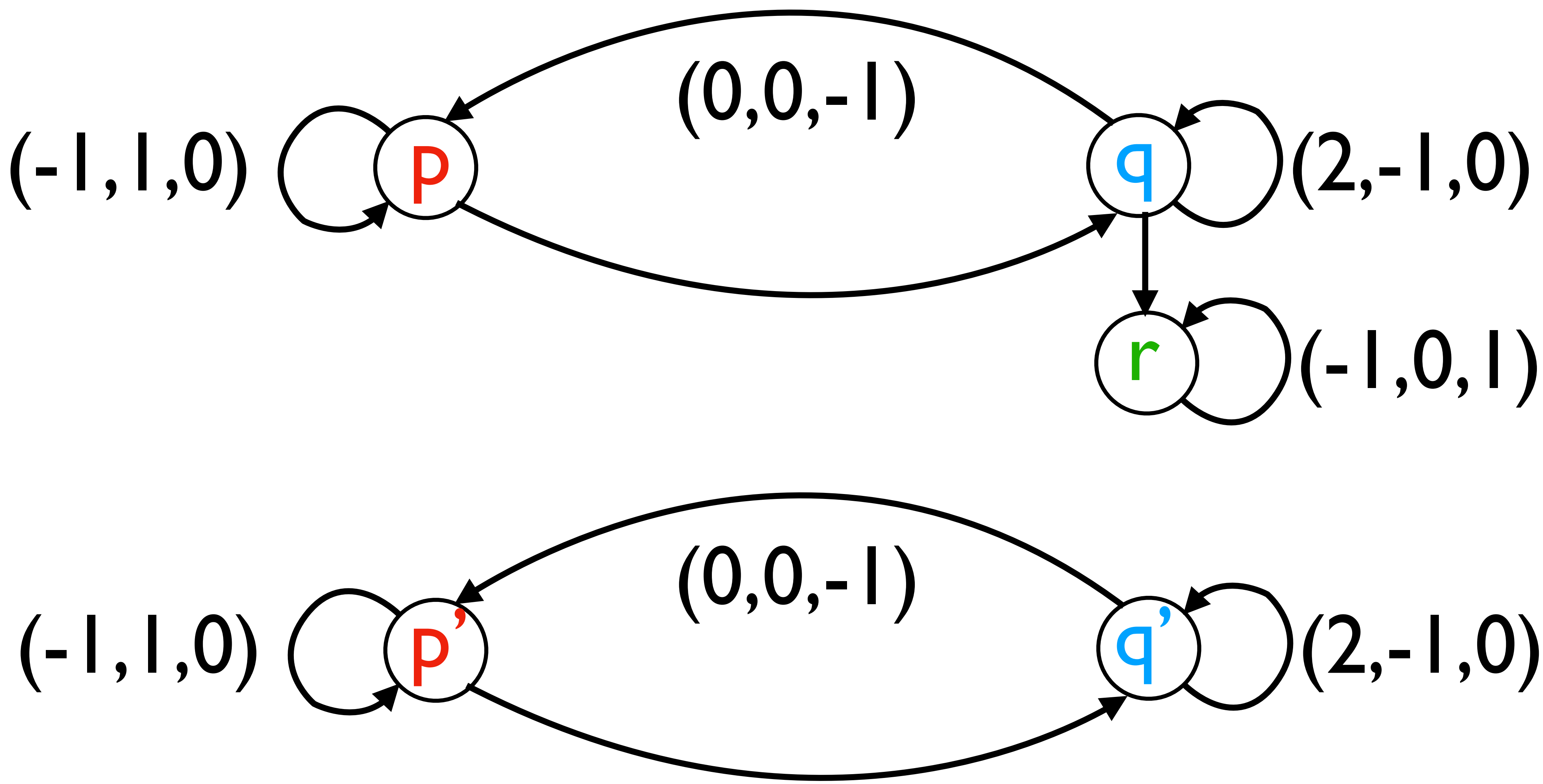




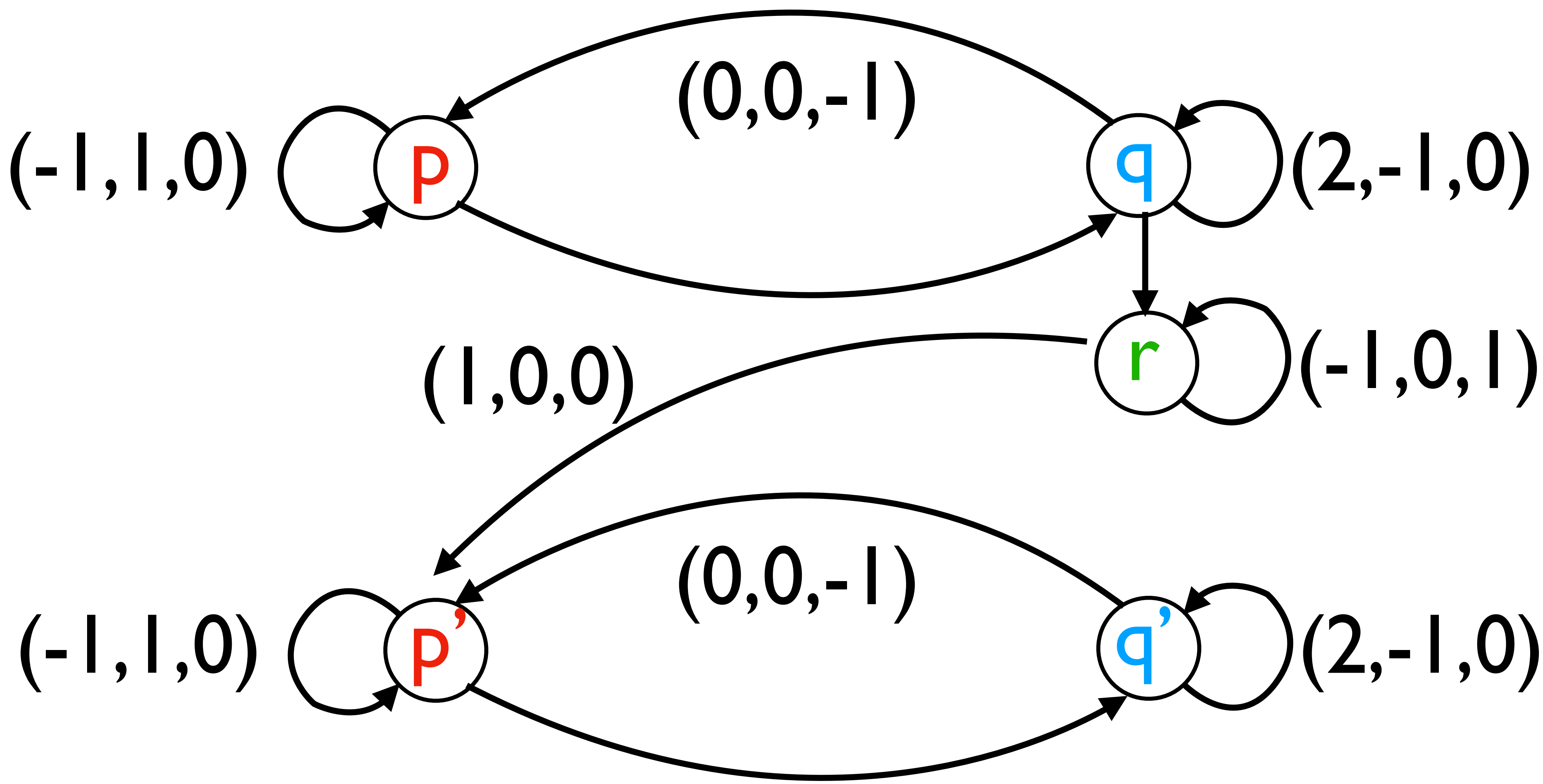
Hard examples



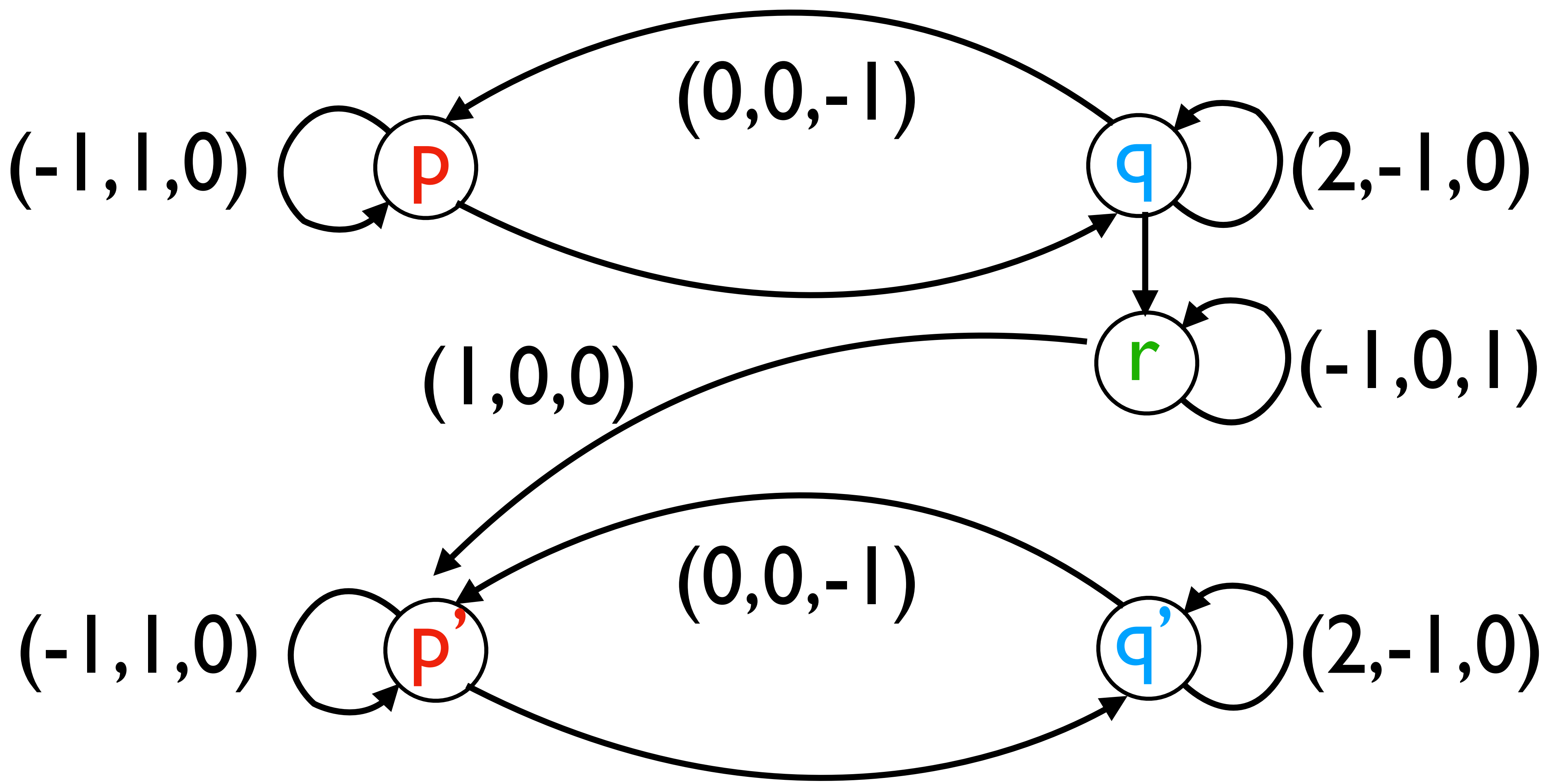
Hard examples



Hard examples

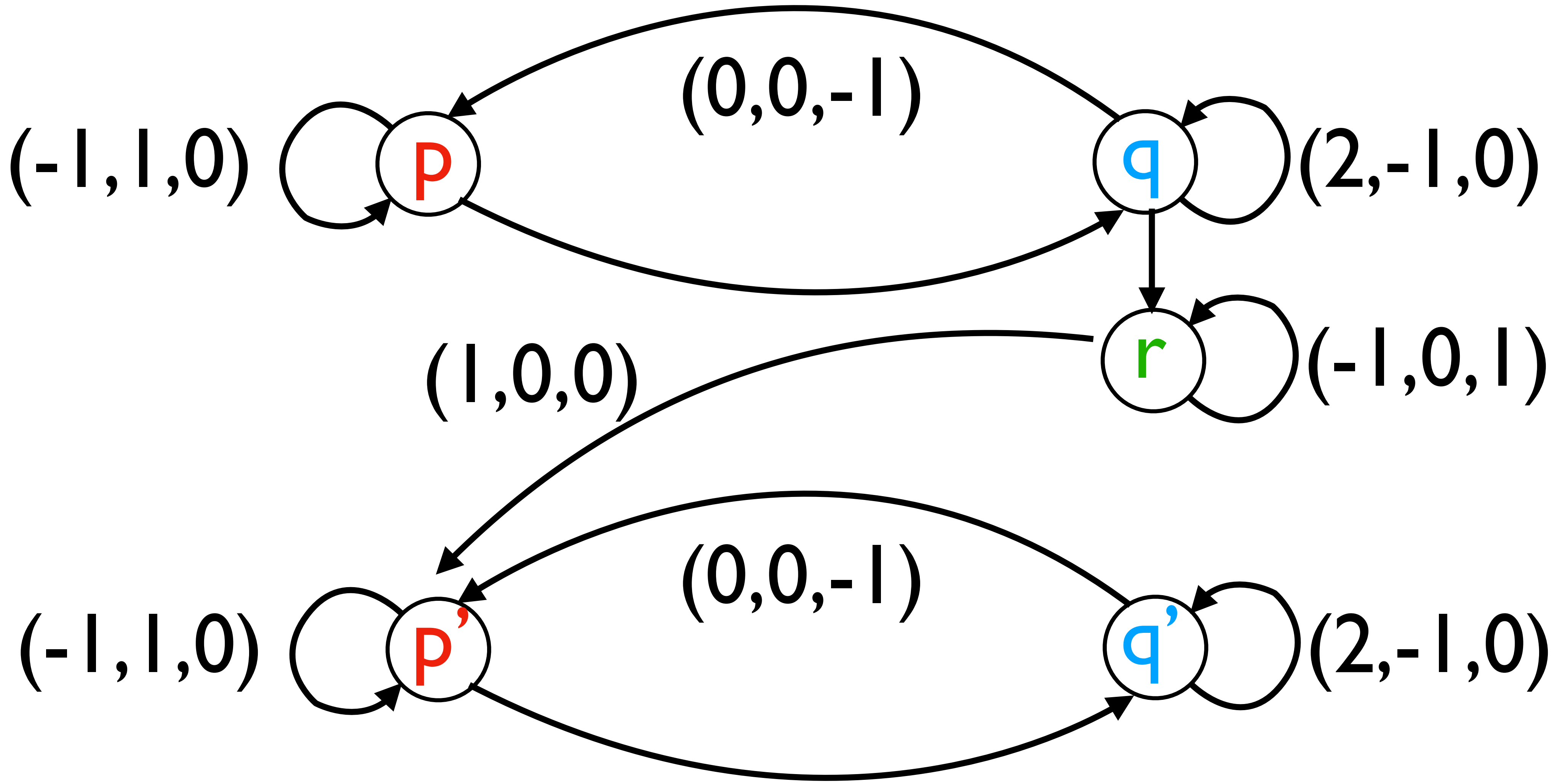


Hard examples



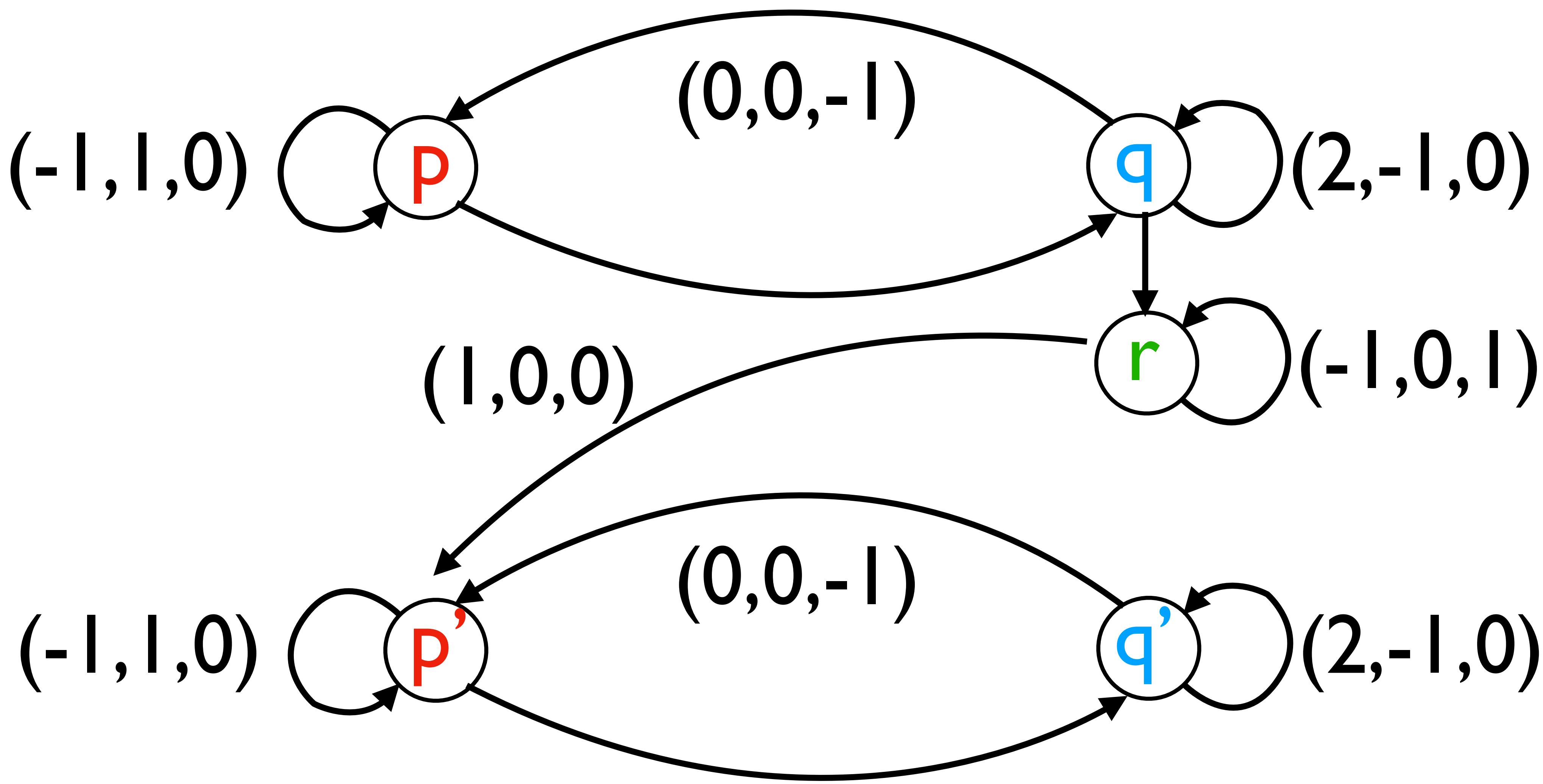
p(1,0,n)

Hard examples



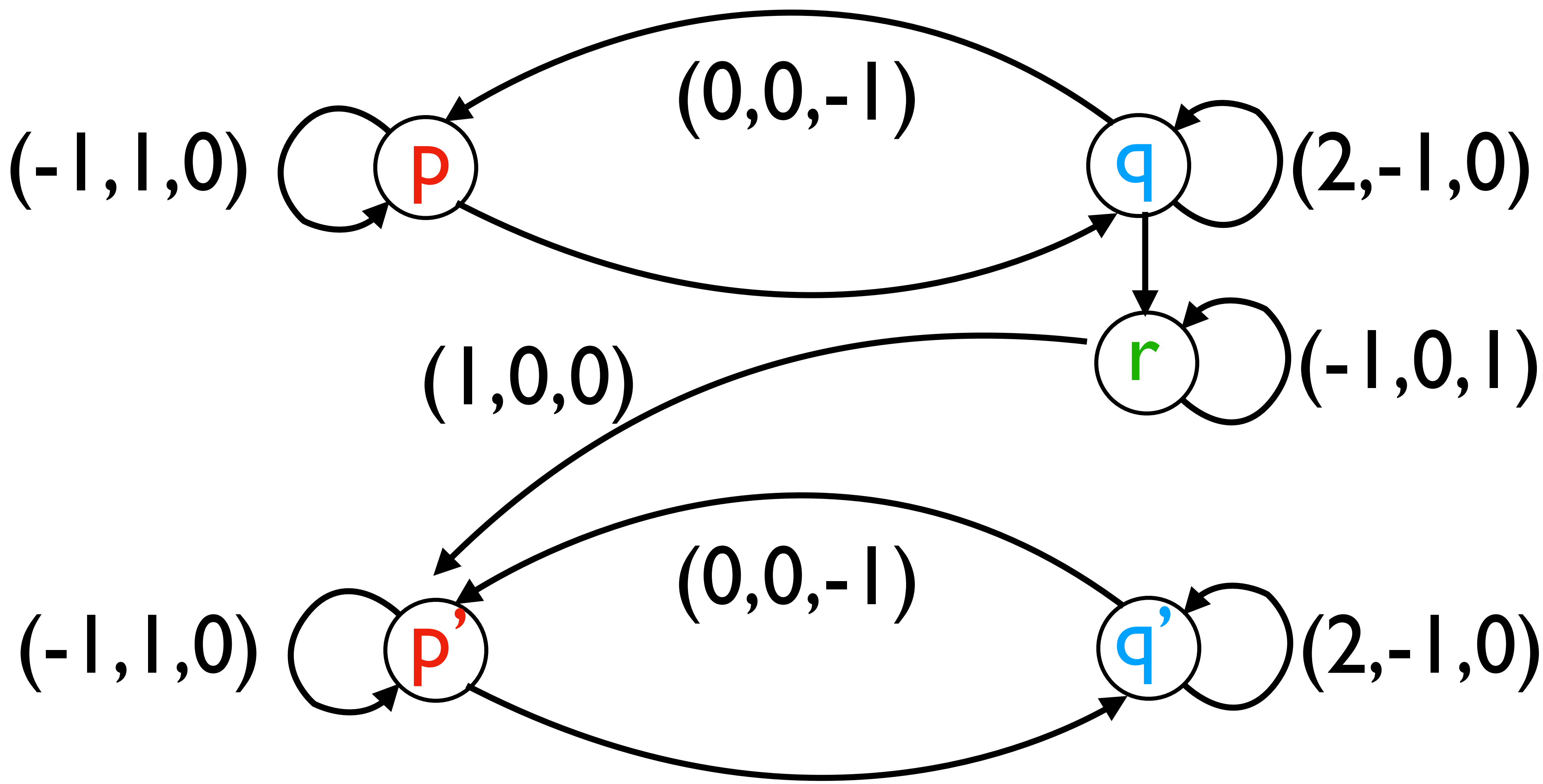
p(1,0,n) ⟶ q(2^n,0,0)

Hard examples



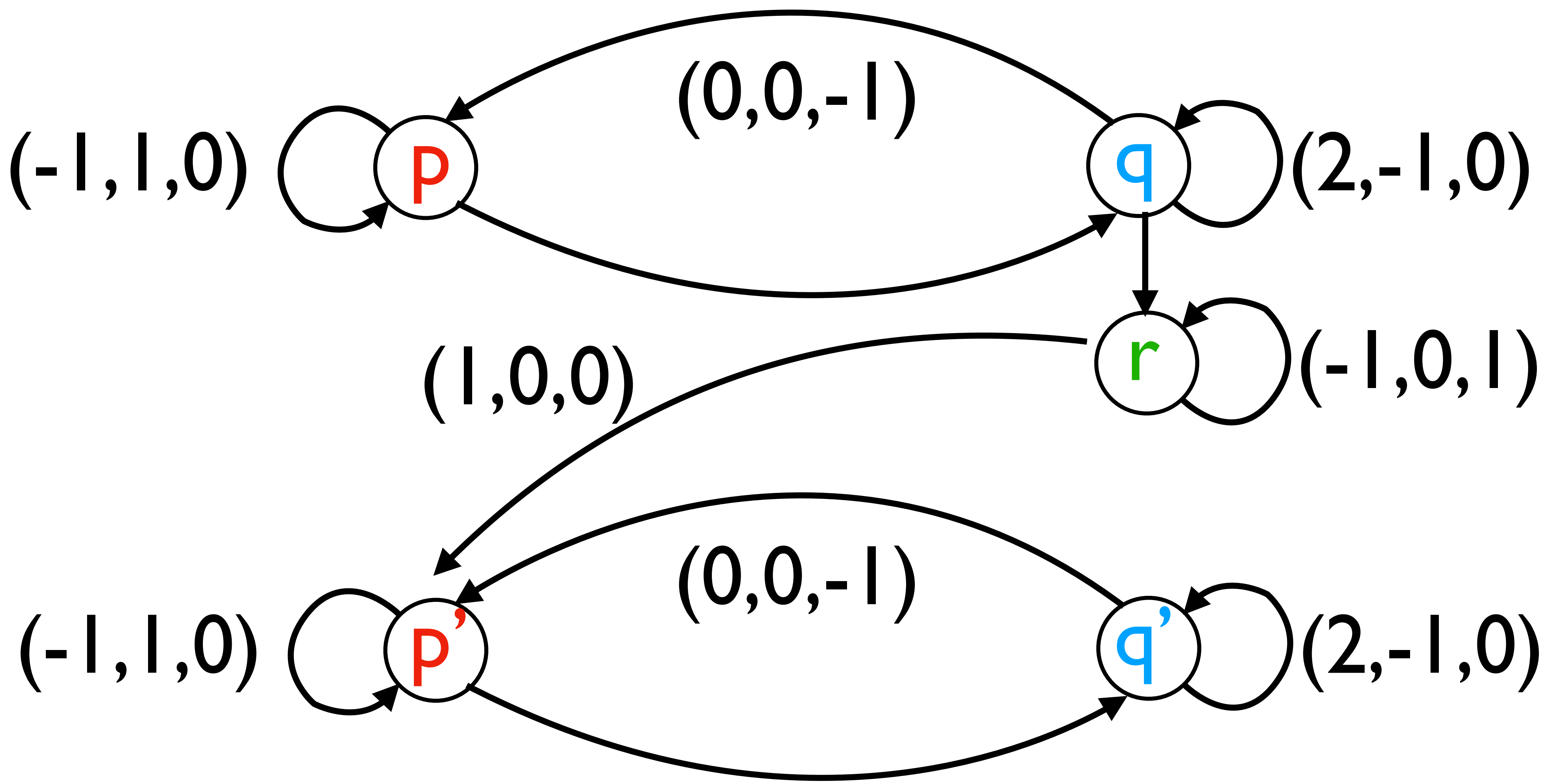
$p(1,0,n) \longrightarrow q(2^n,0,0) \longrightarrow r(2^n,0,0)$

Hard examples



$p(1,0,n) \longrightarrow q(2^n,0,0) \longrightarrow r(2^n,0,0) \longrightarrow r(0,0,2^n)$

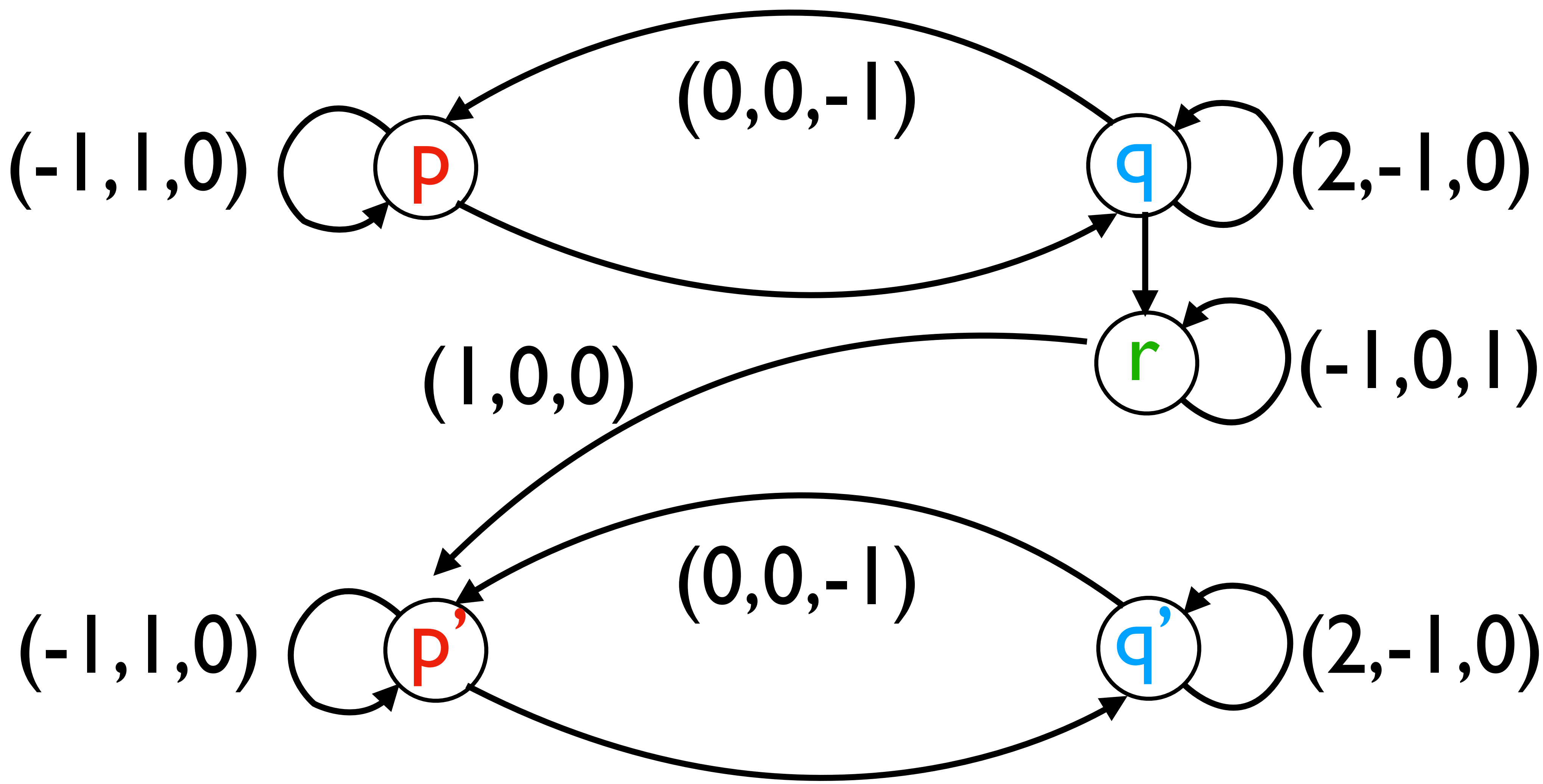
Hard examples



$p(1,0,n) \longrightarrow q(2^n,0,0) \longrightarrow r(2^n,0,0) \longrightarrow r(0,0,2^n)$

$\longrightarrow p'(1,0,2^n)$

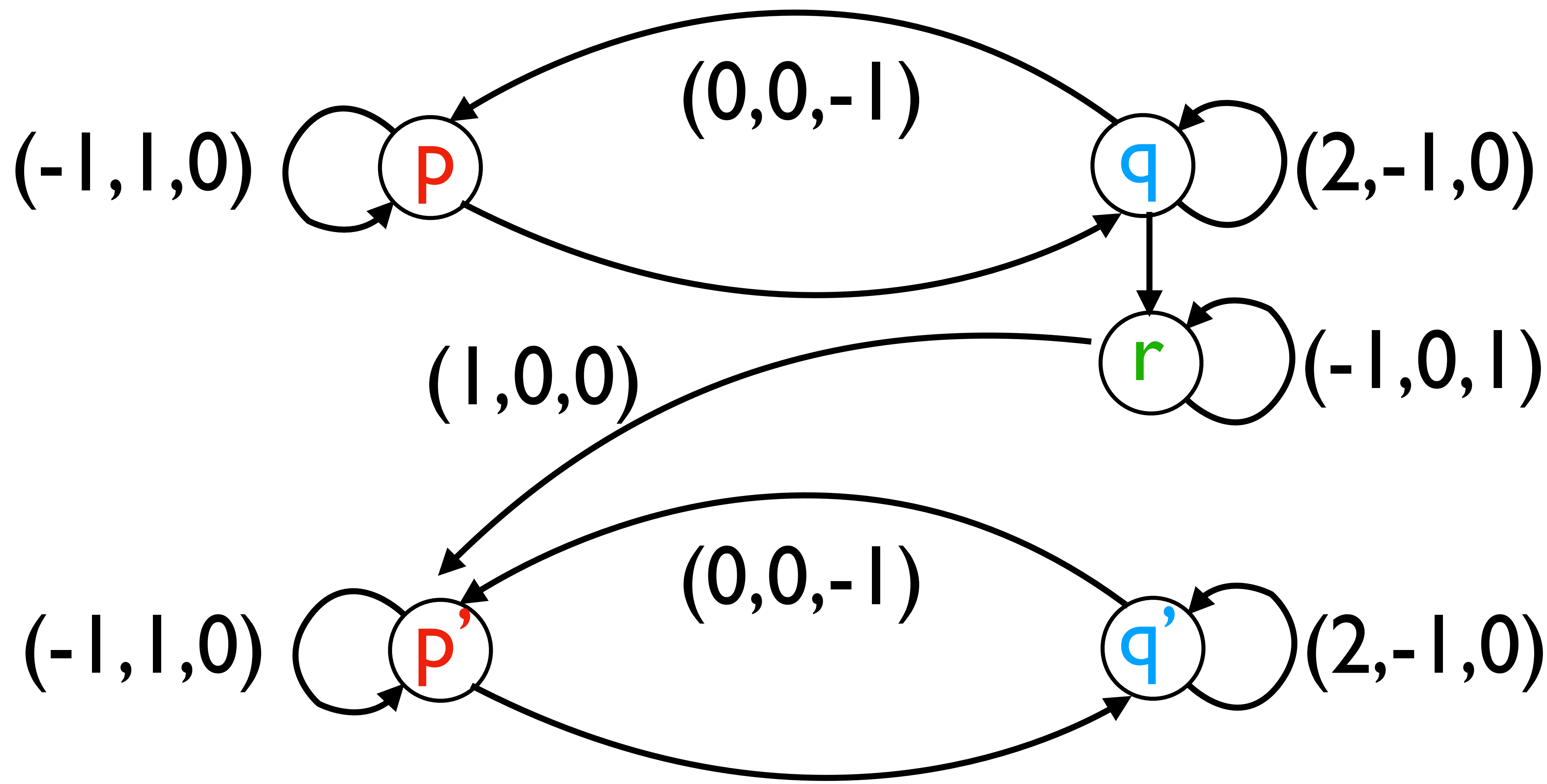
Hard examples



$$p(1,0,n) \longrightarrow q(2^n,0,0) \longrightarrow r(2^n,0,0) \longrightarrow r(0,0,2^n)$$

$$\longrightarrow p'(1,0,2^n) \longrightarrow q'(2^{2^n},0,0)$$

Hard examples



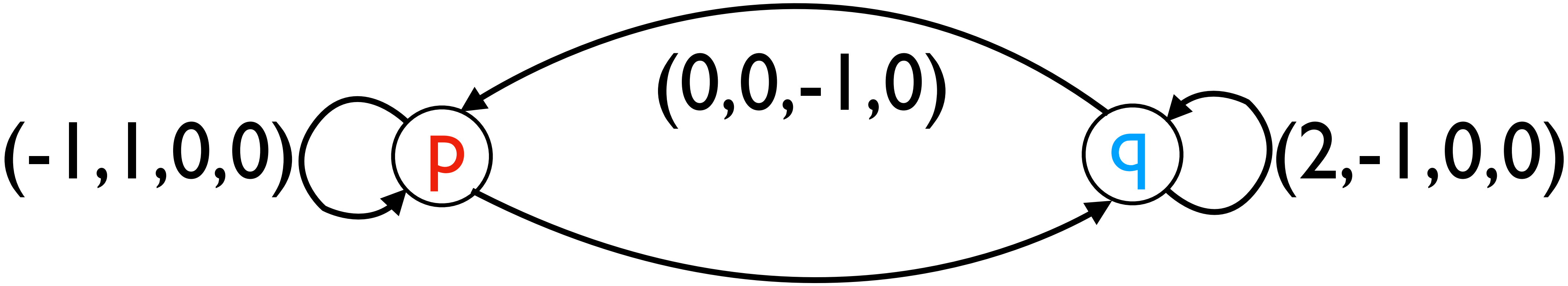
$$p(1,0,n) \longrightarrow q(2^n,0,0) \longrightarrow r(2^n,0,0) \longrightarrow r(0,0,2^n)$$

$$\longrightarrow p'(1,0,2^n) \longrightarrow q'(2^{2^n},0,0)$$

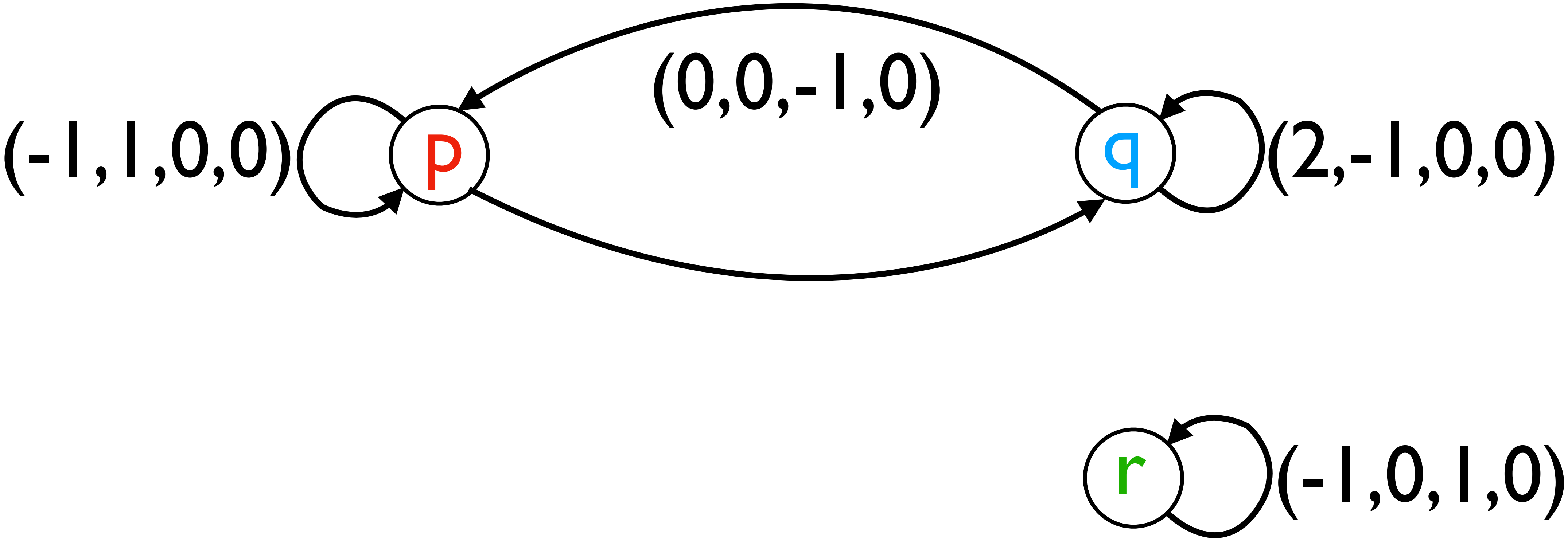
finite doubly-exponential reachability set

Hard examples

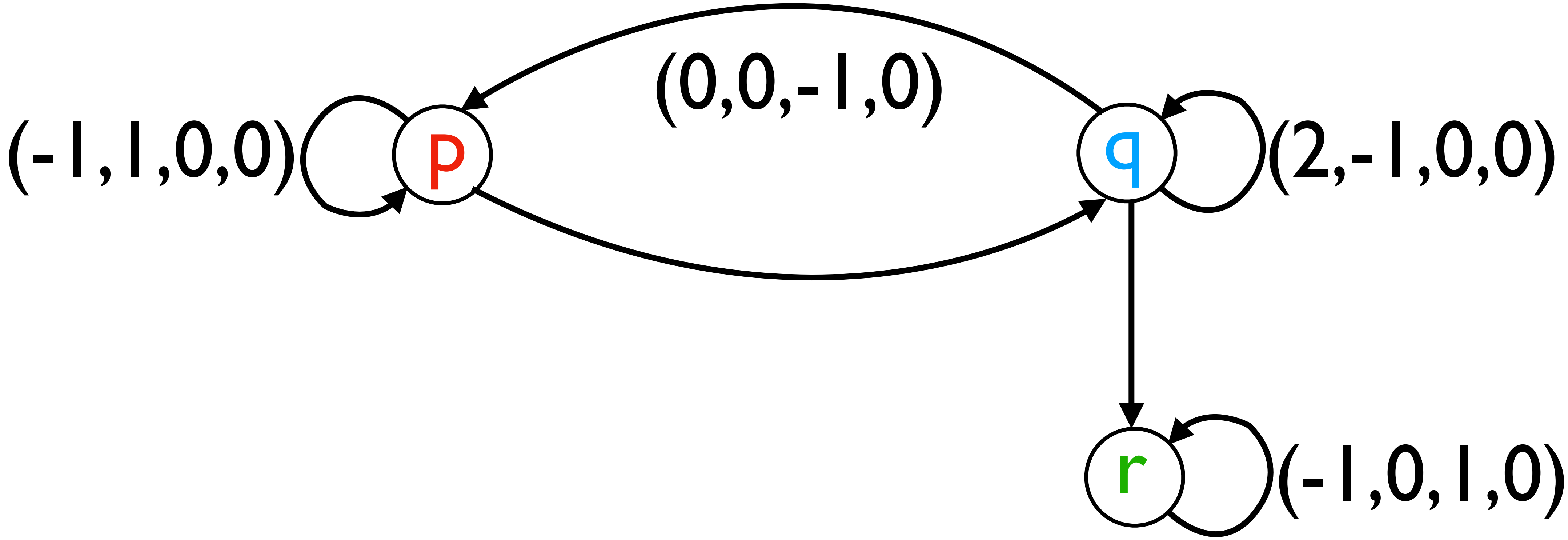
Hard examples



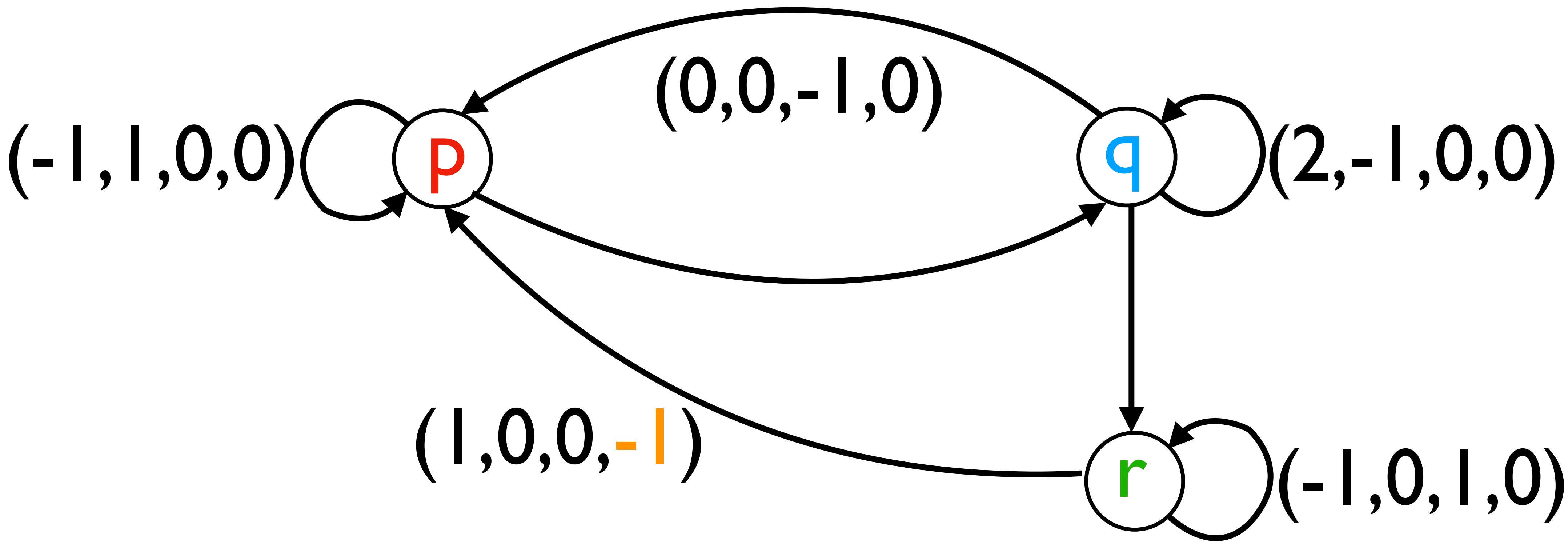
Hard examples



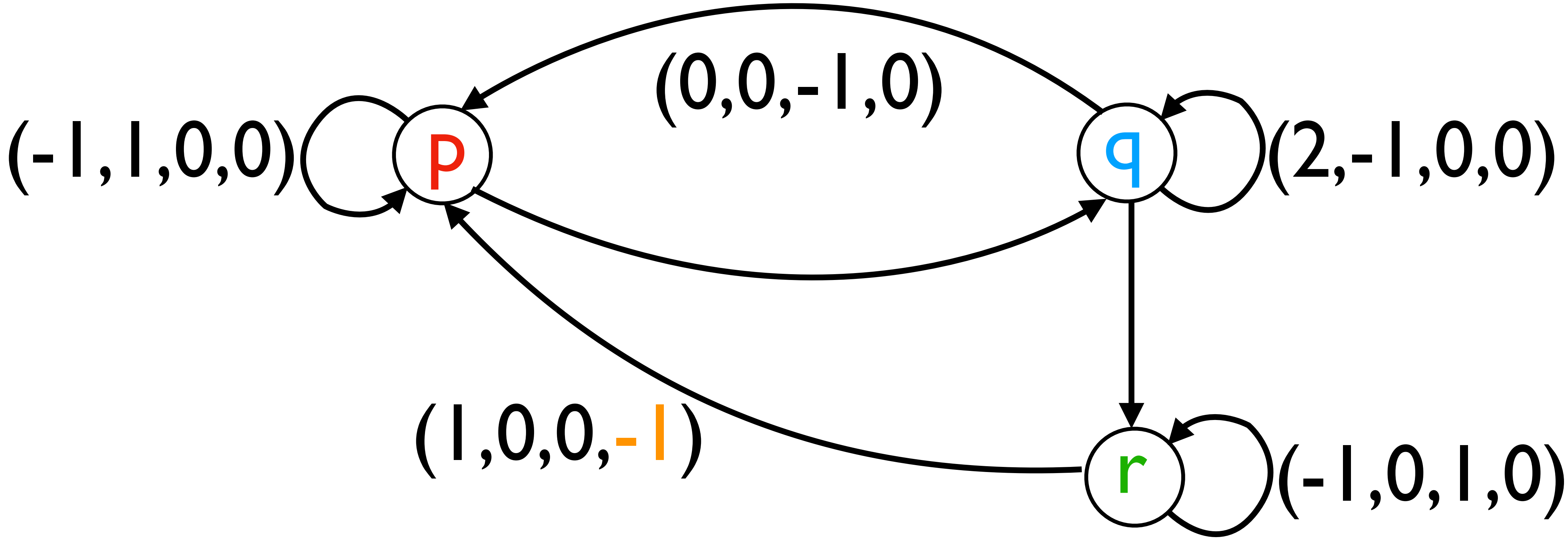
Hard examples



Hard examples

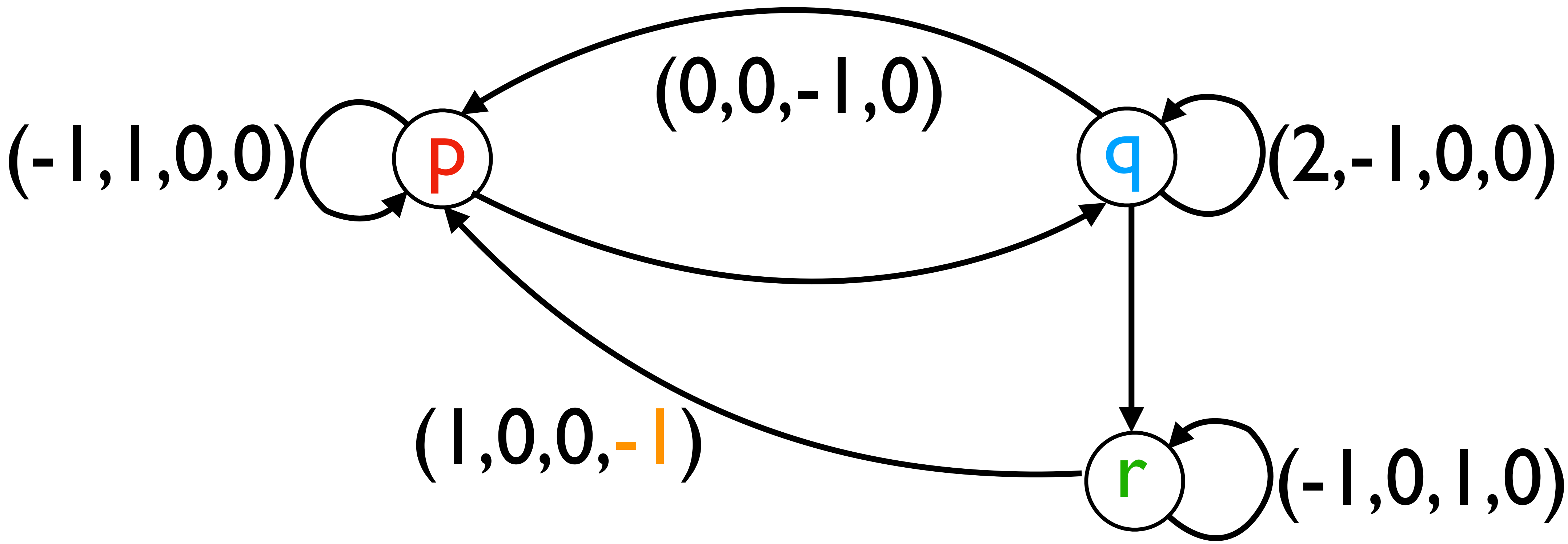


Hard examples



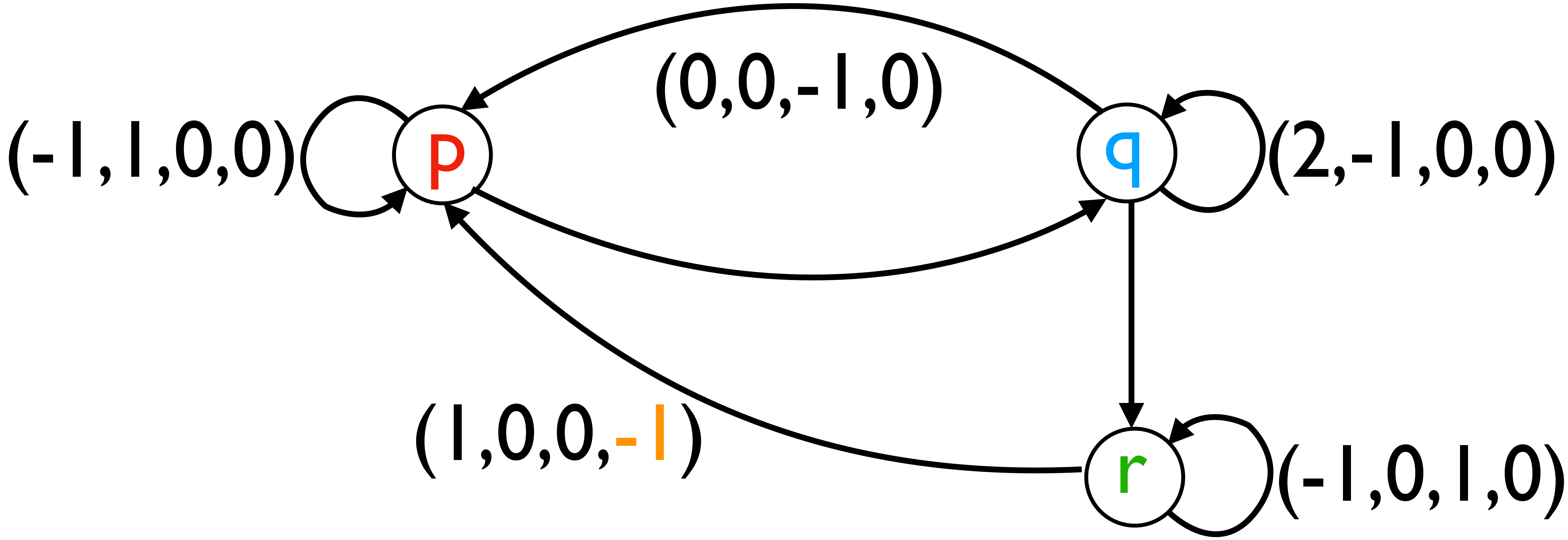
p(1,0,1,n)

Hard examples



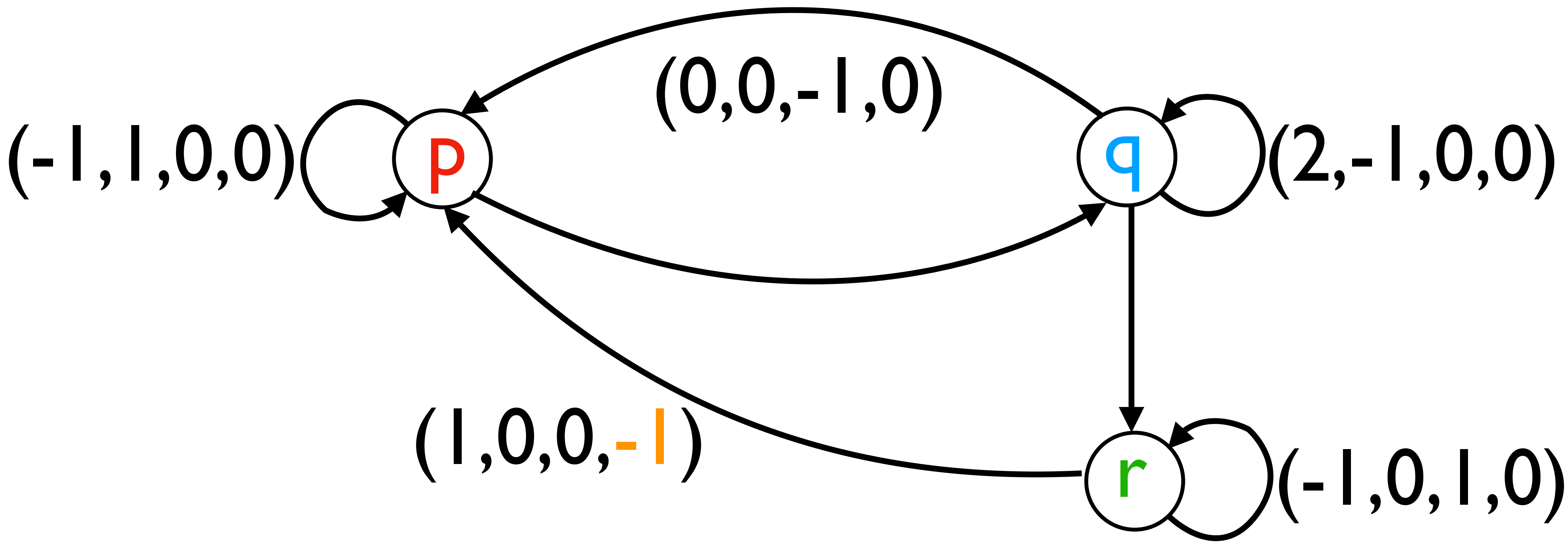
p(1,0,1,n) ⟶ p(2^{1},0,1,n-1)

Hard examples



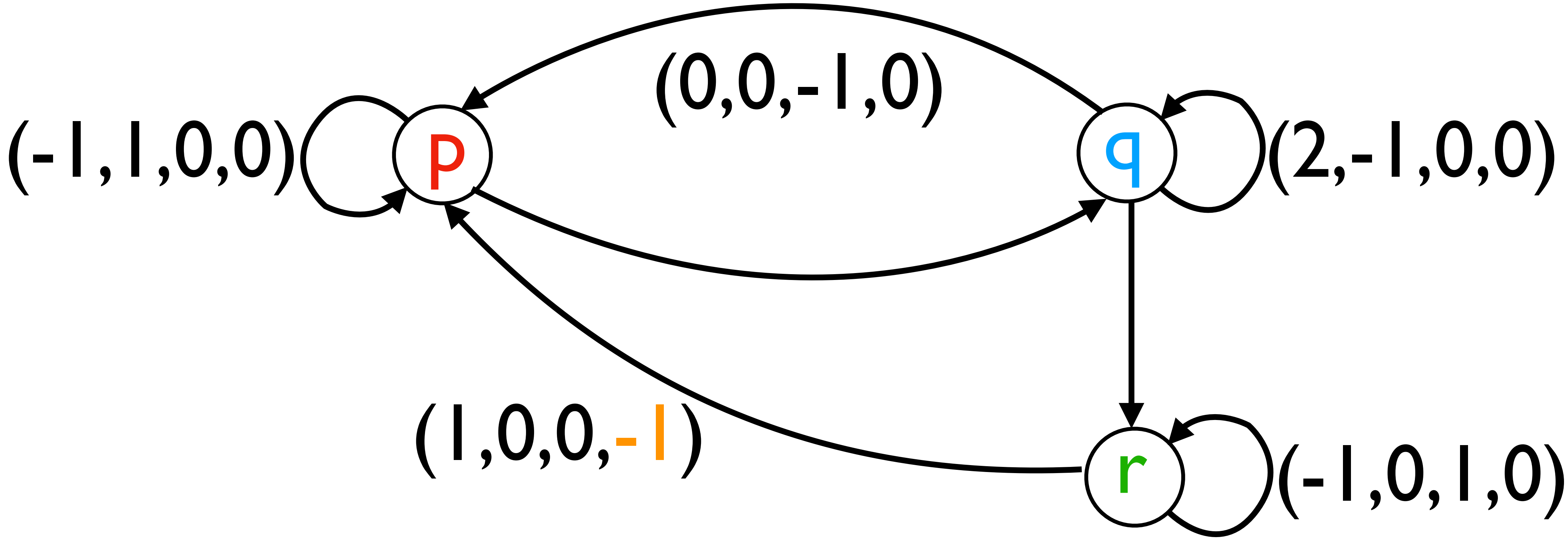
$p(1,0,1,n) \longrightarrow p(2^1,0,1,n-1) \ldots$

Hard examples



$p(1,0,1,n) \longrightarrow p(2^1,0,1,n-1) \ldots \longrightarrow p(\text{Tower}(n),0,1,0)$

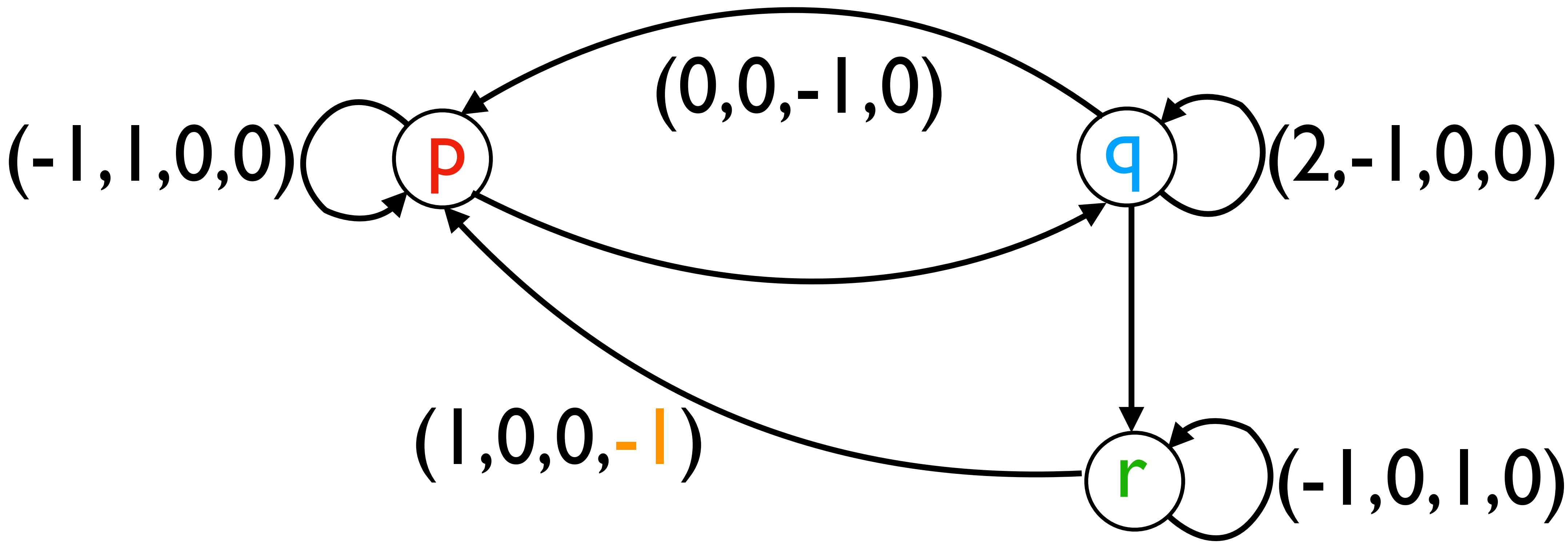
Hard examples



p(1,0,1,n) $\longrightarrow$ p(2^1,0,1,n-1) … $\longrightarrow$ p(Tower(n),0,1,0)

finite tower-size reachability set

Hard examples



p(1,0,1,n) ⟶ p(2^1,0,1,n-1) … ⟶ p(Tower(n),0,1,0)

finite tower-size reachability set

finite F_d-size reachability set

Decidability idea

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Decidability idea

Question: does p(s) $\longrightarrow$ q(t)?

Idea: reachability in $\mathbb{Z}$ is easy!

Decidability idea

Question: does p(s) $\longrightarrow$ q(t)?

Idea: reachability in $\mathbb{Z}$ is easy!

Linear equations (in NP)

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume:

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume: $p(s) \longrightarrow p(s+\Delta)$

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume: $p(s) \longrightarrow p(s+\Delta)$ $q(t+\Delta) \longrightarrow q(t)$

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume: $p(s) \longrightarrow p(s+\Delta)$ $\quad q(t+\Delta) \longrightarrow q(t)$

with $\Delta \geq 1$

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume: $p(s) \longrightarrow p(s+\Delta)$ $q(t+\Delta) \longrightarrow q(t)$

with $\Delta \geq 1$

Then:

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume: $p(s) \longrightarrow p(s+\Delta)$ $\quad q(t+\Delta) \longrightarrow q(t)$

with $\Delta \geq 1$

Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume: $p(s) \longrightarrow p(s+\Delta)$ $\quad q(t+\Delta) \longrightarrow q(t)$

with $\Delta \geq 1$

Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$ implies

Decidability idea

Question: does $p(s) \longrightarrow q(t)$?

Idea: reachability in $\mathbb{Z}$ is easy!

Assume: $p(s) \longrightarrow p(s+\Delta)$ $\quad q(t+\Delta) \longrightarrow q(t)$

with $\Delta \geq 1$

Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$ implies $p(s) \longrightarrow q(t)$

Decidability idea

Decidability idea

Then:

Decidability idea

Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$

Decidability idea

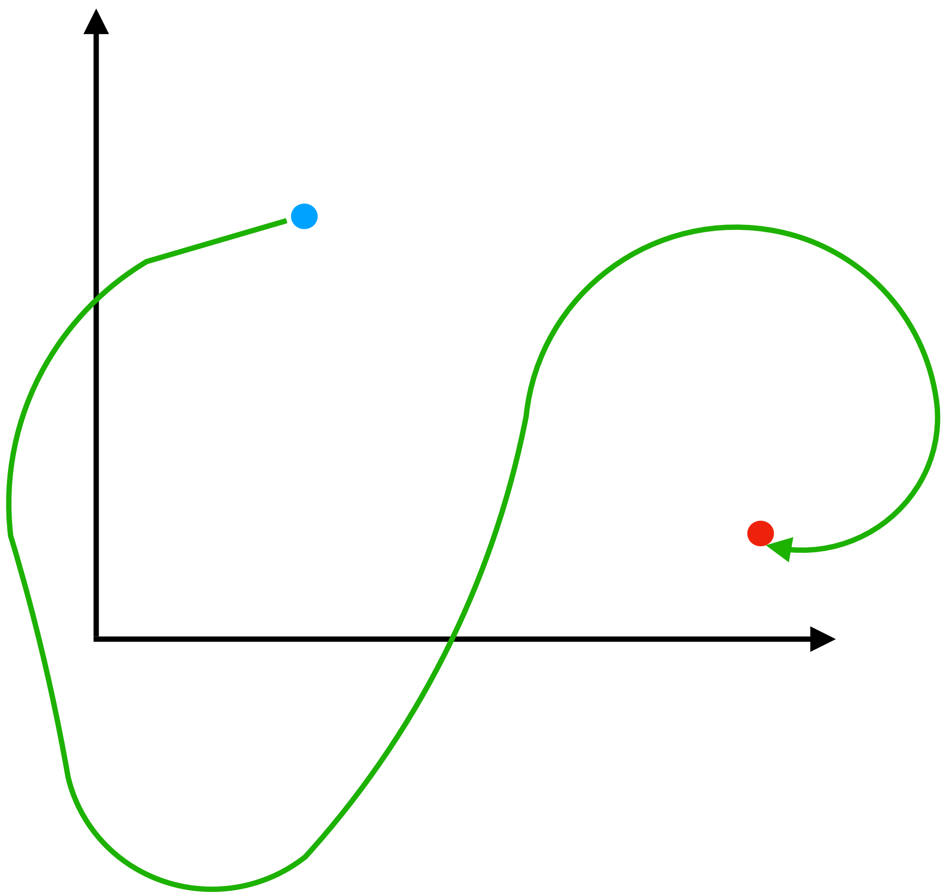
Then: p(s) $\xrightarrow{\mathbb{Z}}$ q(t) implies

Decidability idea

Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$ implies $p(s) \longrightarrow q(t)$

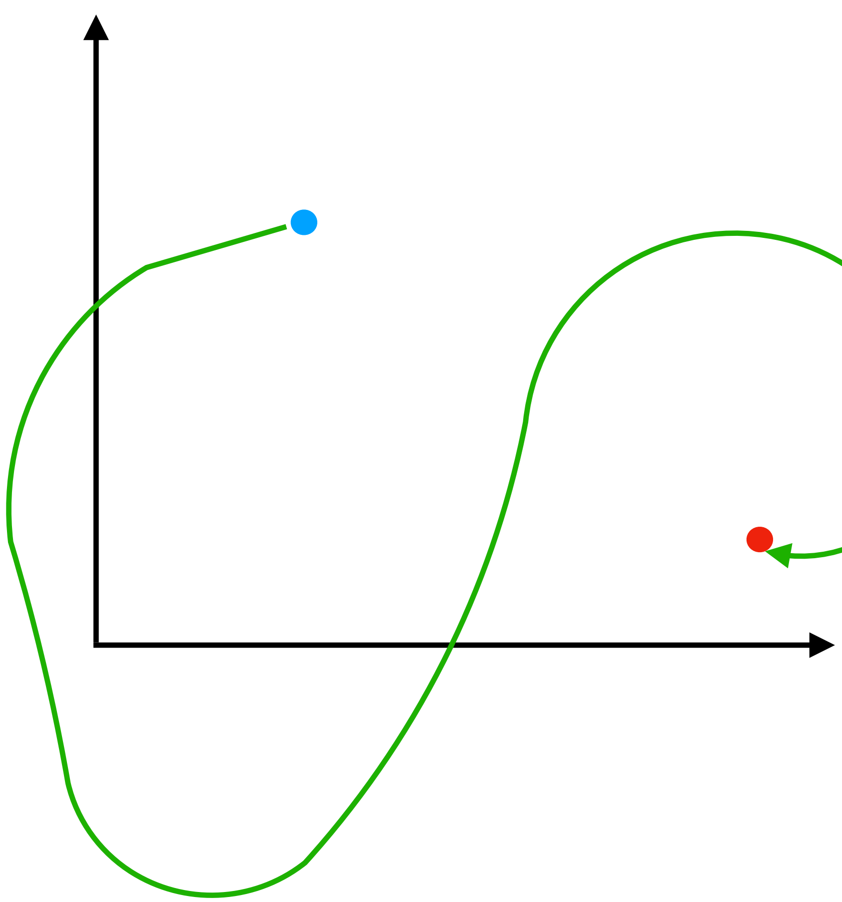
Decidability idea

Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$ implies $p(s) \longrightarrow q(t)$



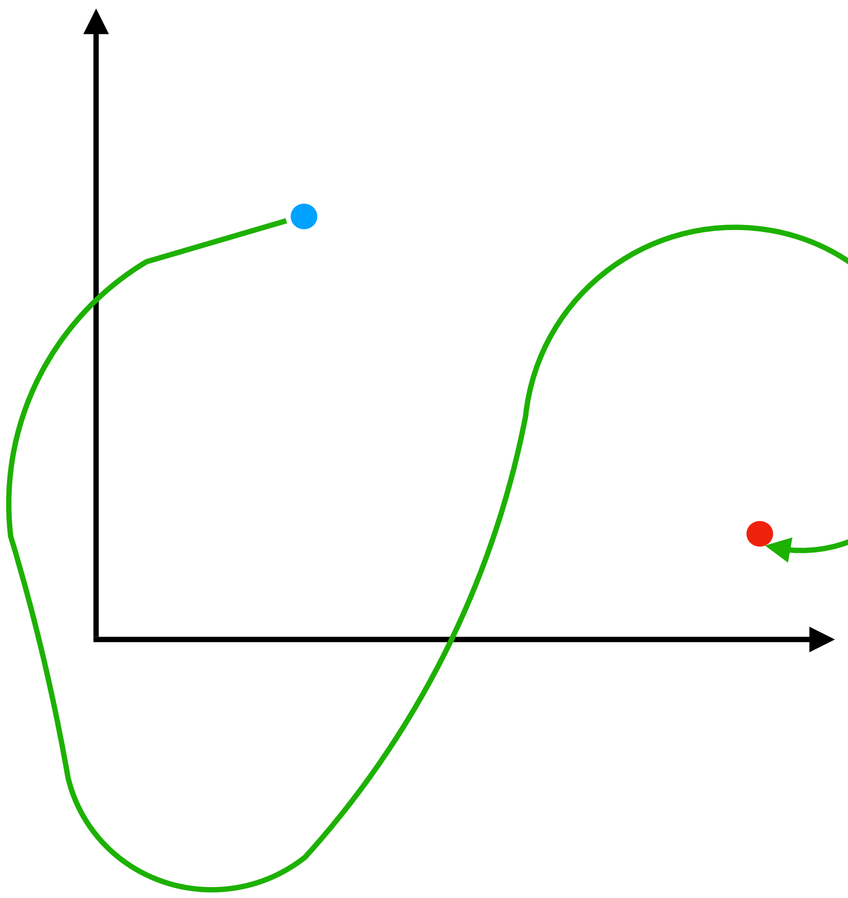
Decidability idea

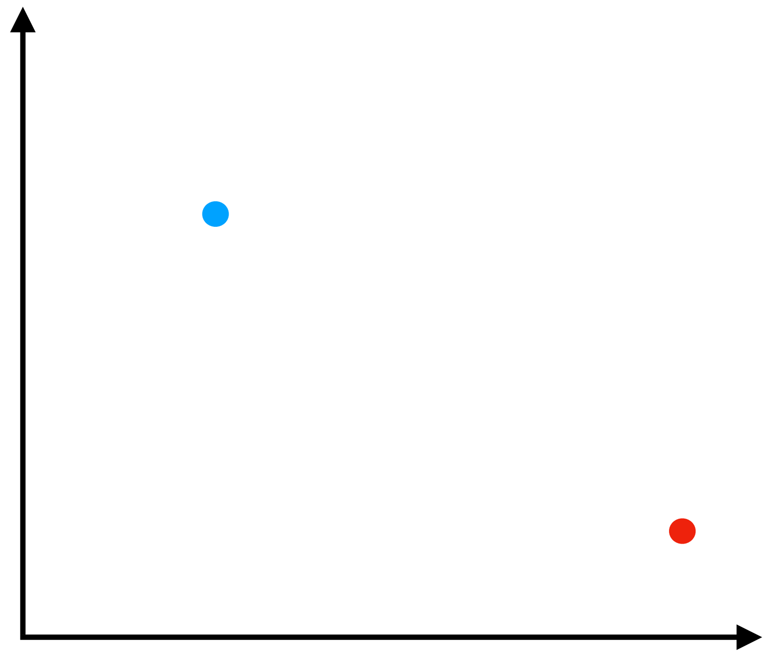
Then: p(s) $\xrightarrow{\mathbb{Z}}$ q(t) implies p(s) $\longrightarrow$ q(t)



Decidability idea

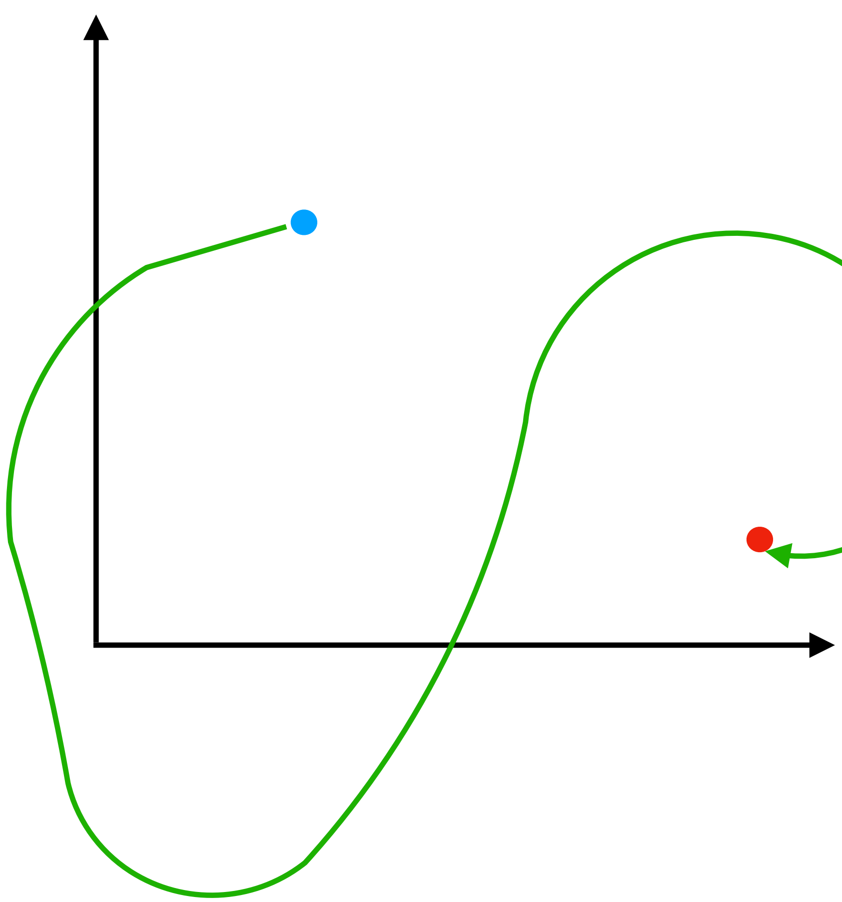
Then: p(s) $\xrightarrow{\mathbb{Z}}$ q(t) implies p(s) $\longrightarrow$ q(t)

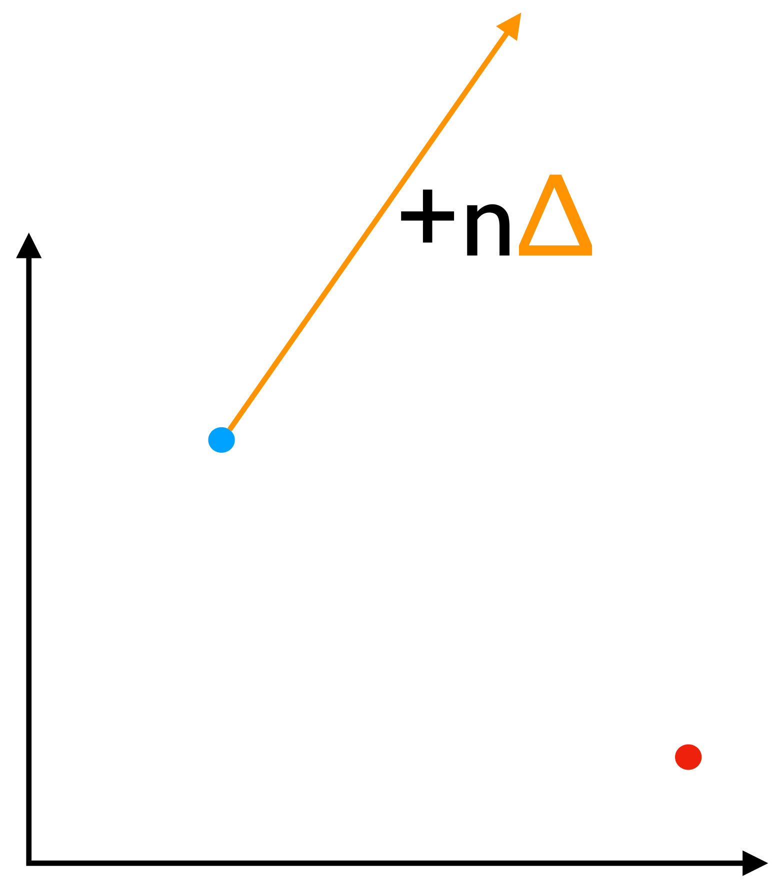




Decidability idea

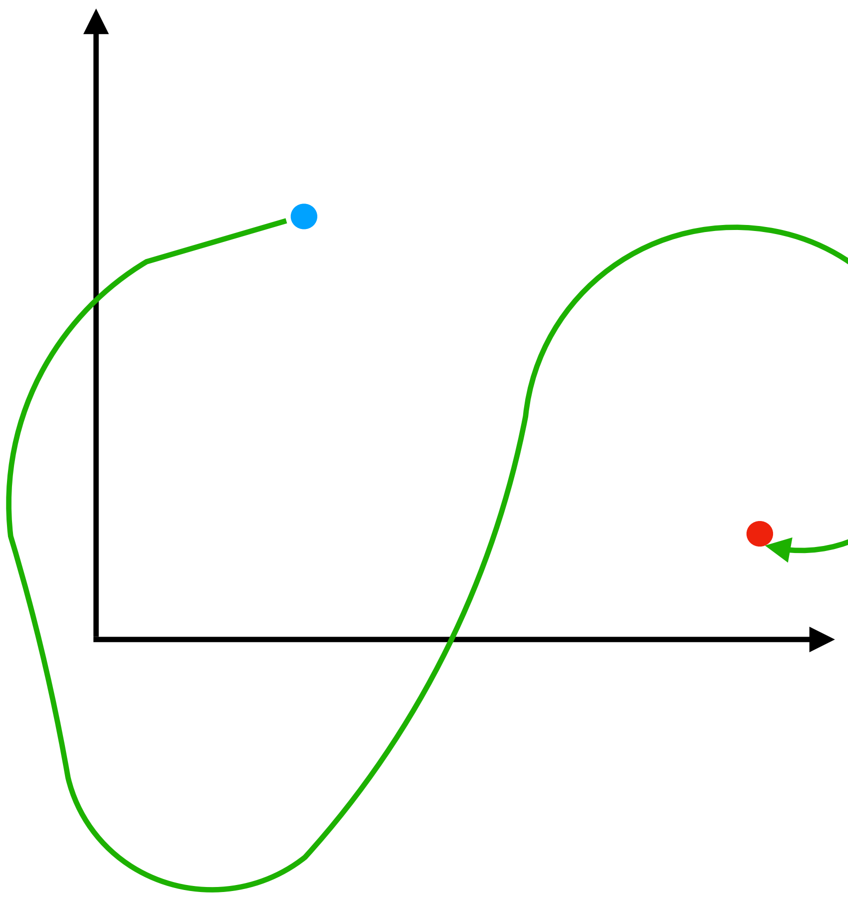
Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$ implies $p(s) \longrightarrow q(t)$

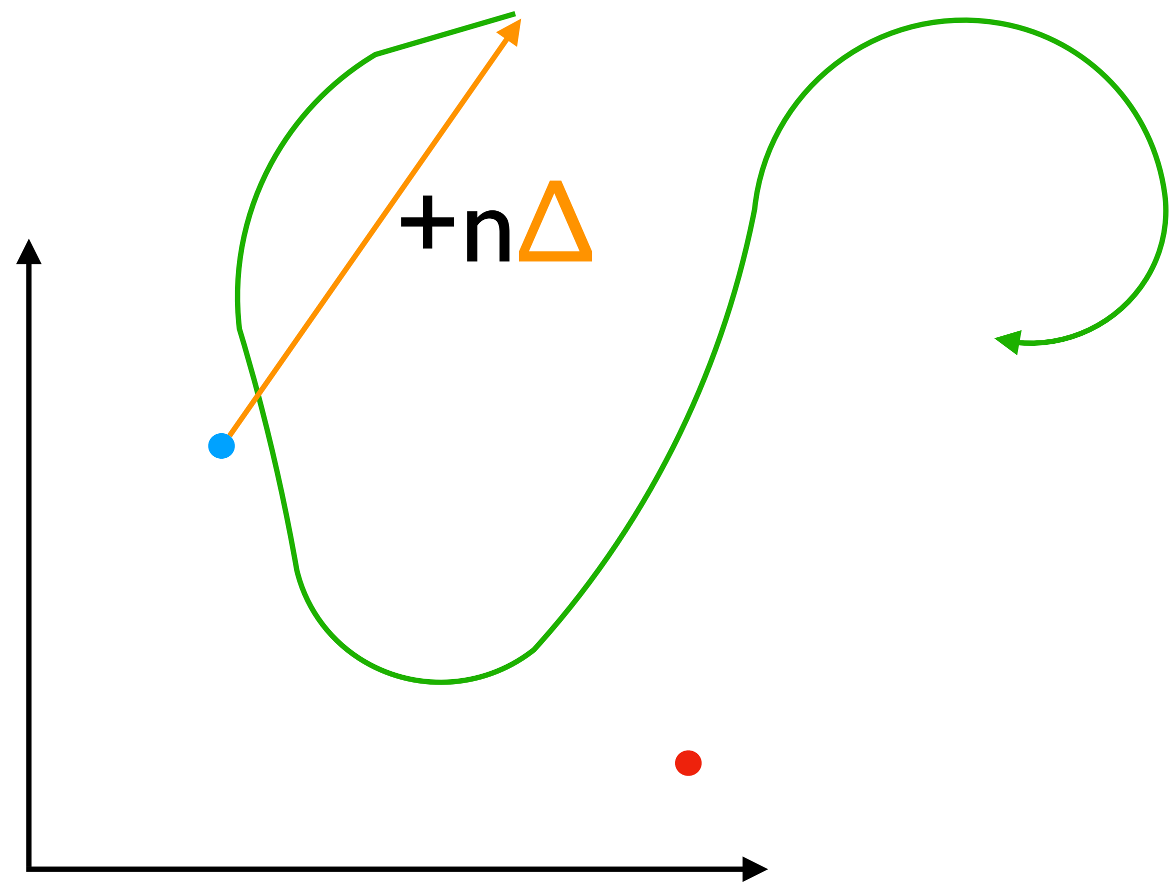




Decidability idea

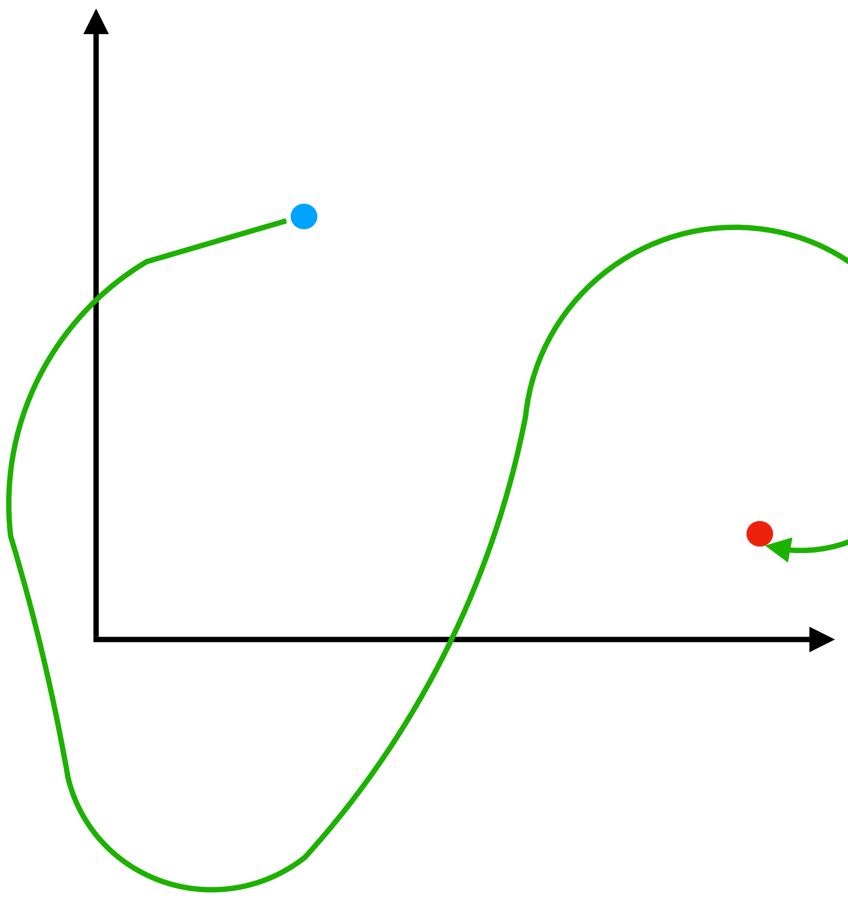
Then: $p(s) \xrightarrow{\mathbb{Z}} q(t)$ implies $p(s) \longrightarrow q(t)$

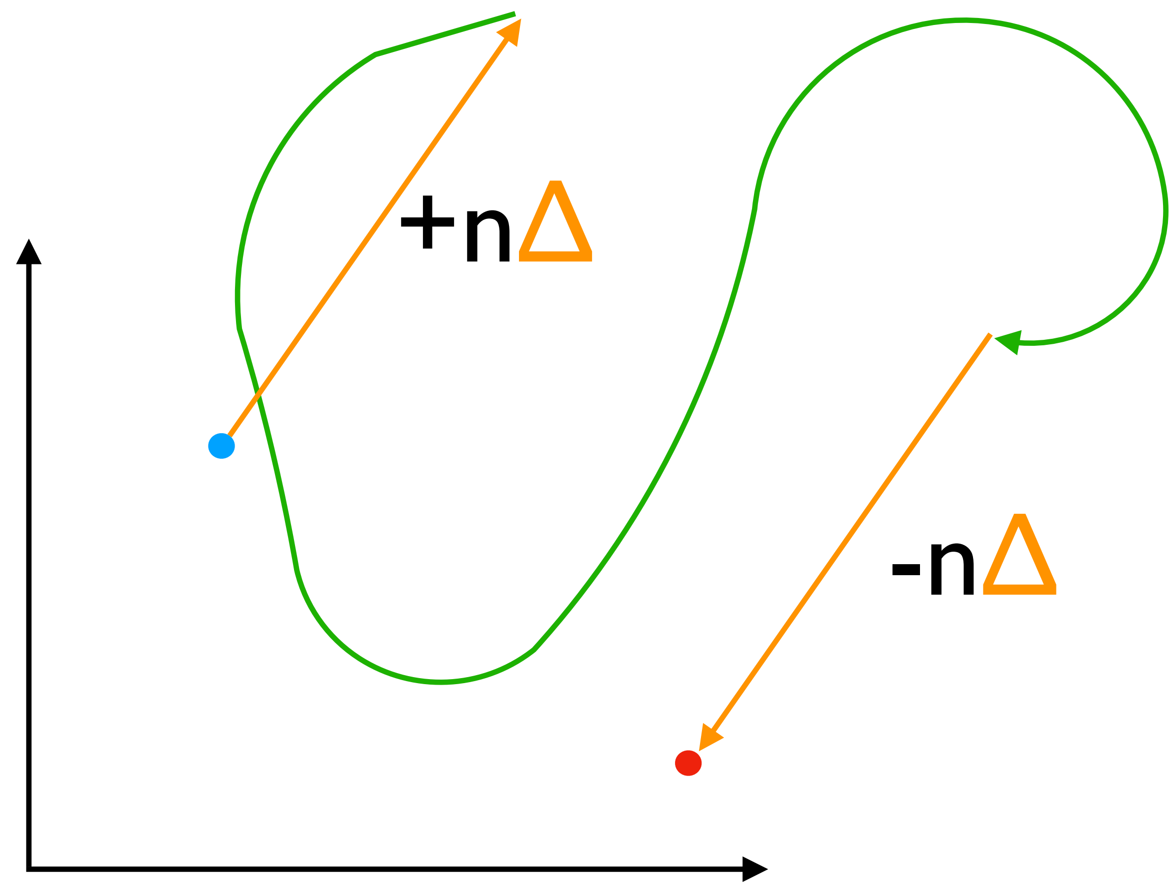




Decidability idea

Then: p(s) $\xrightarrow{\mathbb{Z}}$ q(t) implies p(s) $\longrightarrow$ q(t)





Decidability idea

Decidability idea

Assume:

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1)$

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1)$ $q(t+\Delta_2) \longrightarrow q(t)$

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1)$ $q(t+\Delta_2) \longrightarrow q(t)$

Then:

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1)$ $\quad q(t+\Delta_2) \longrightarrow q(t)$

Then:

$p(s) \xrightarrow{\mathbb{Z}} q(t)$

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1)$ $\quad q(t+\Delta_2) \longrightarrow q(t)$

Then:

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1) \quad q(t+\Delta_2) \longrightarrow q(t)$

Then:

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

implies

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1)$ $q(t+\Delta_2) \longrightarrow q(t)$

Then:

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

implies $p(s) \longrightarrow q(t)$

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1) \quad q(t+\Delta_2) \longrightarrow q(t)$

Then:

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

implies $p(s) \longrightarrow q(t)$

Why?

Decidability idea

Assume: $p(s) \longrightarrow p(s+\Delta_1)$ $\quad q(t+\Delta_2) \longrightarrow q(t)$

Then:

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

implies $\quad p(s) \longrightarrow q(t)$

Why? $\quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$

Decidability idea

Decidability idea

$p(s) \longrightarrow p(s+\Delta_1)$

Decidability idea

$p(s) \longrightarrow p(s+\Delta_1) \quad q(t+\Delta_2) \longrightarrow q(t)$

Decidability idea

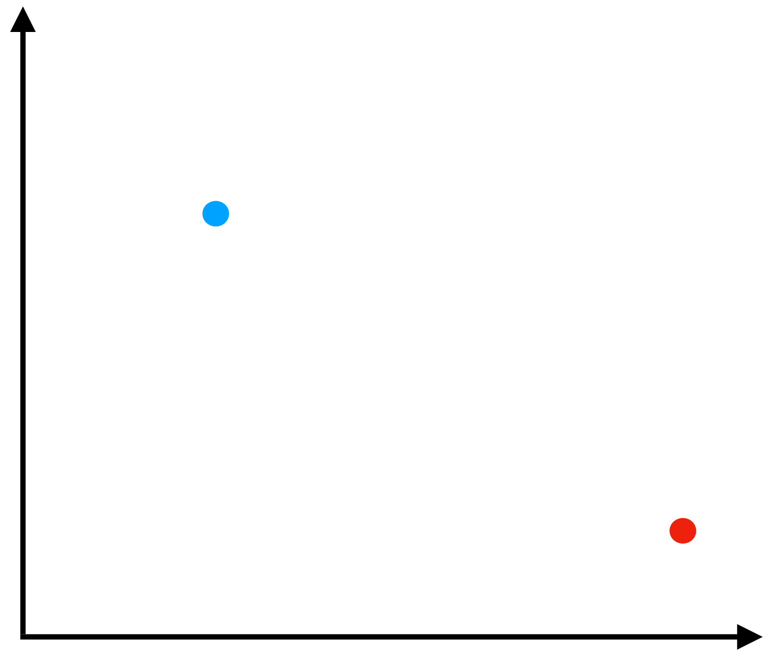
$p(s) \longrightarrow p(s+\Delta_1)$ $\quad q(t+\Delta_2) \longrightarrow q(t)$ $\quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$

Decidability idea

$p(s) \longrightarrow p(s+\Delta_1)$ $\quad q(t+\Delta_2) \longrightarrow q(t)$ $\quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$

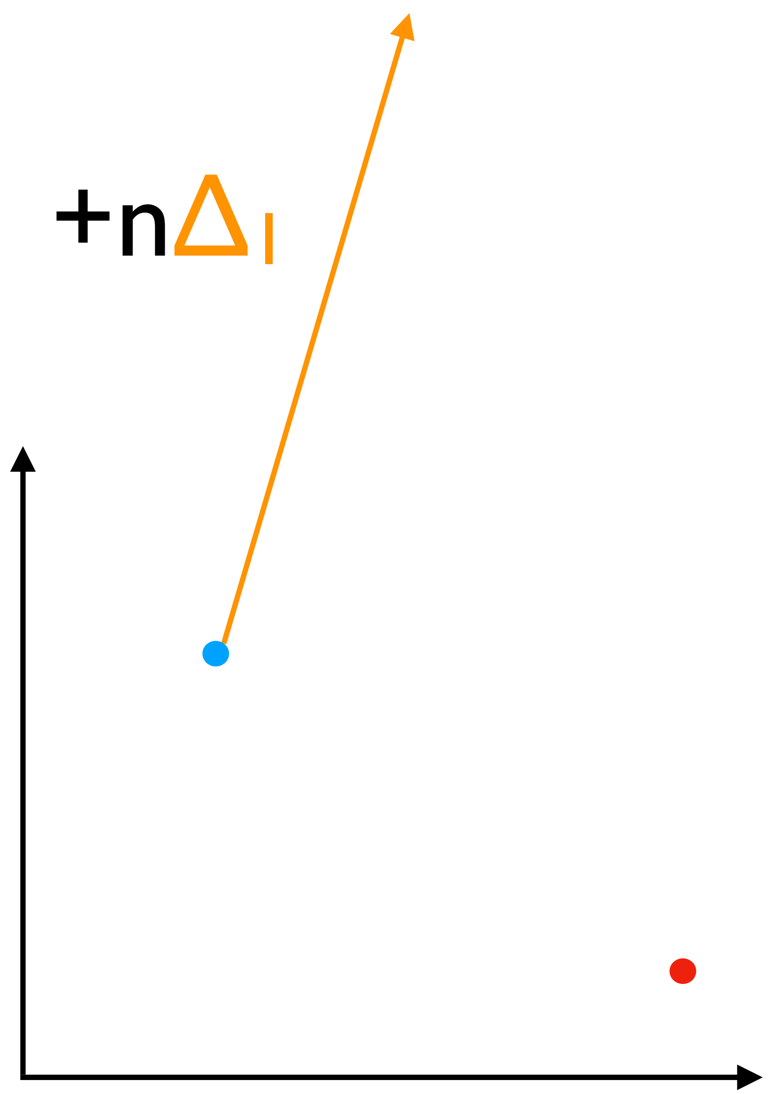
Decidability idea

$p(s) \longrightarrow p(s+\Delta_1) \quad q(t+\Delta_2) \longrightarrow q(t) \quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$



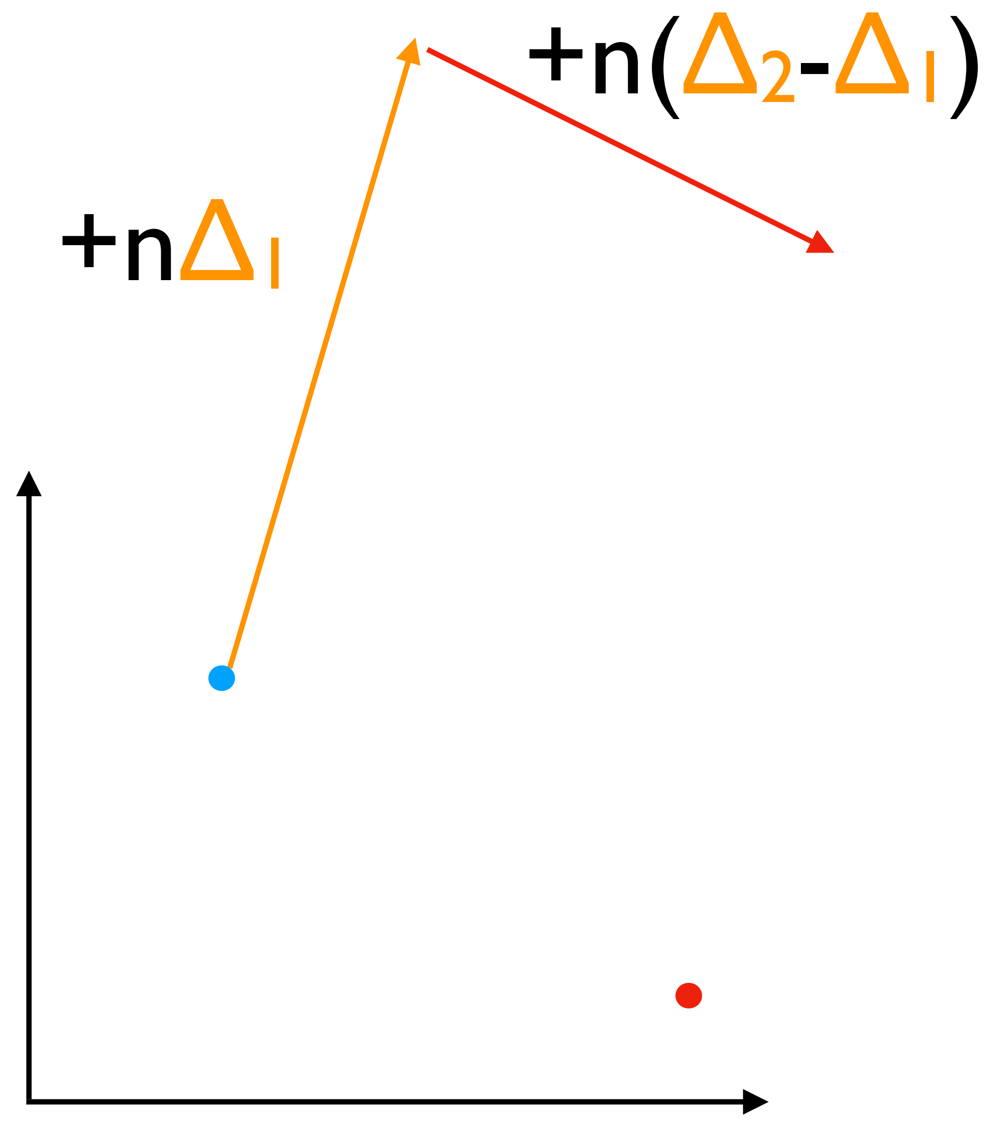
Decidability idea

$p(s) \longrightarrow p(s+\Delta_1) \quad q(t+\Delta_2) \longrightarrow q(t) \quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$



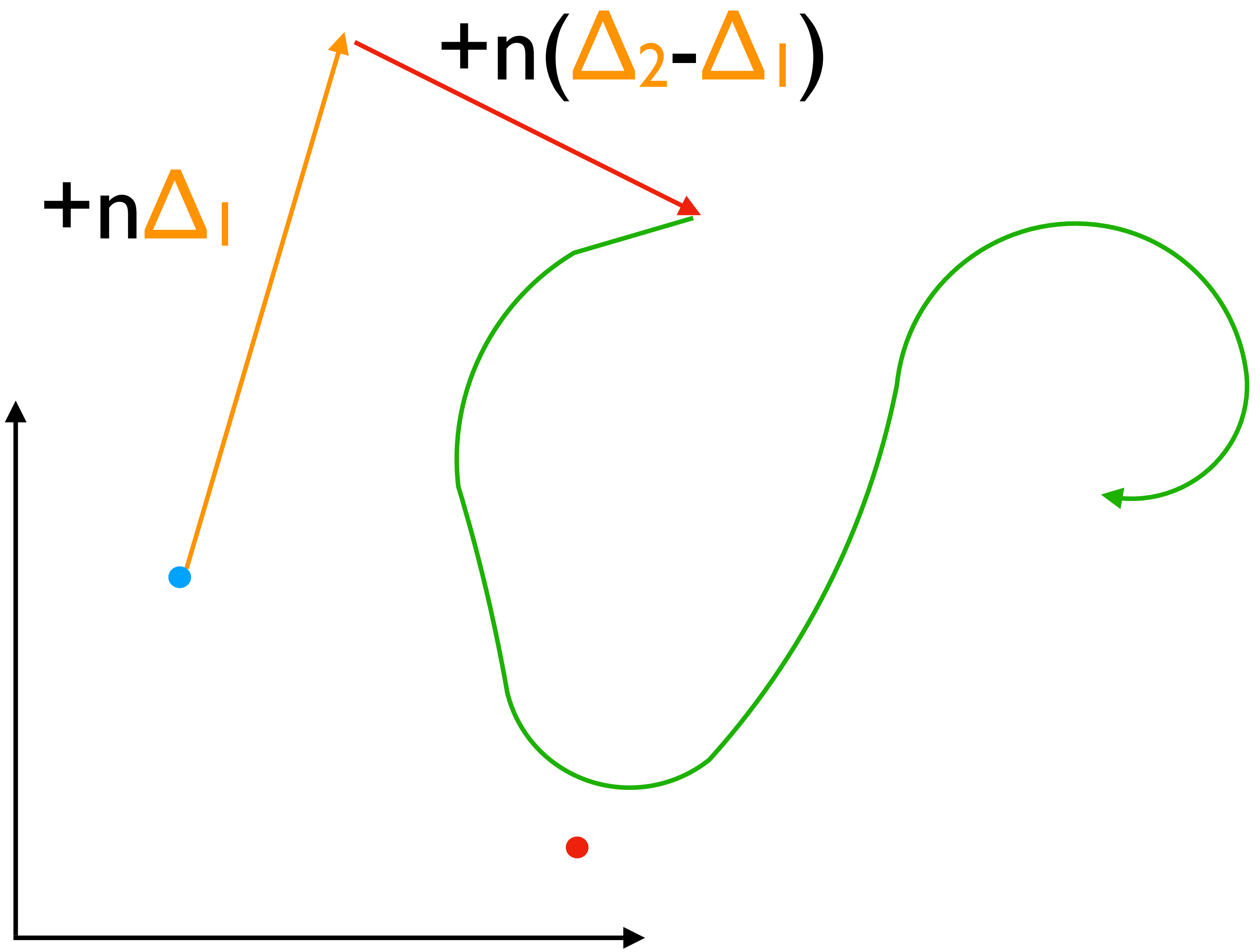
Decidability idea

$p(s) \longrightarrow p(s+\Delta_1) \quad q(t+\Delta_2) \longrightarrow q(t) \quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$



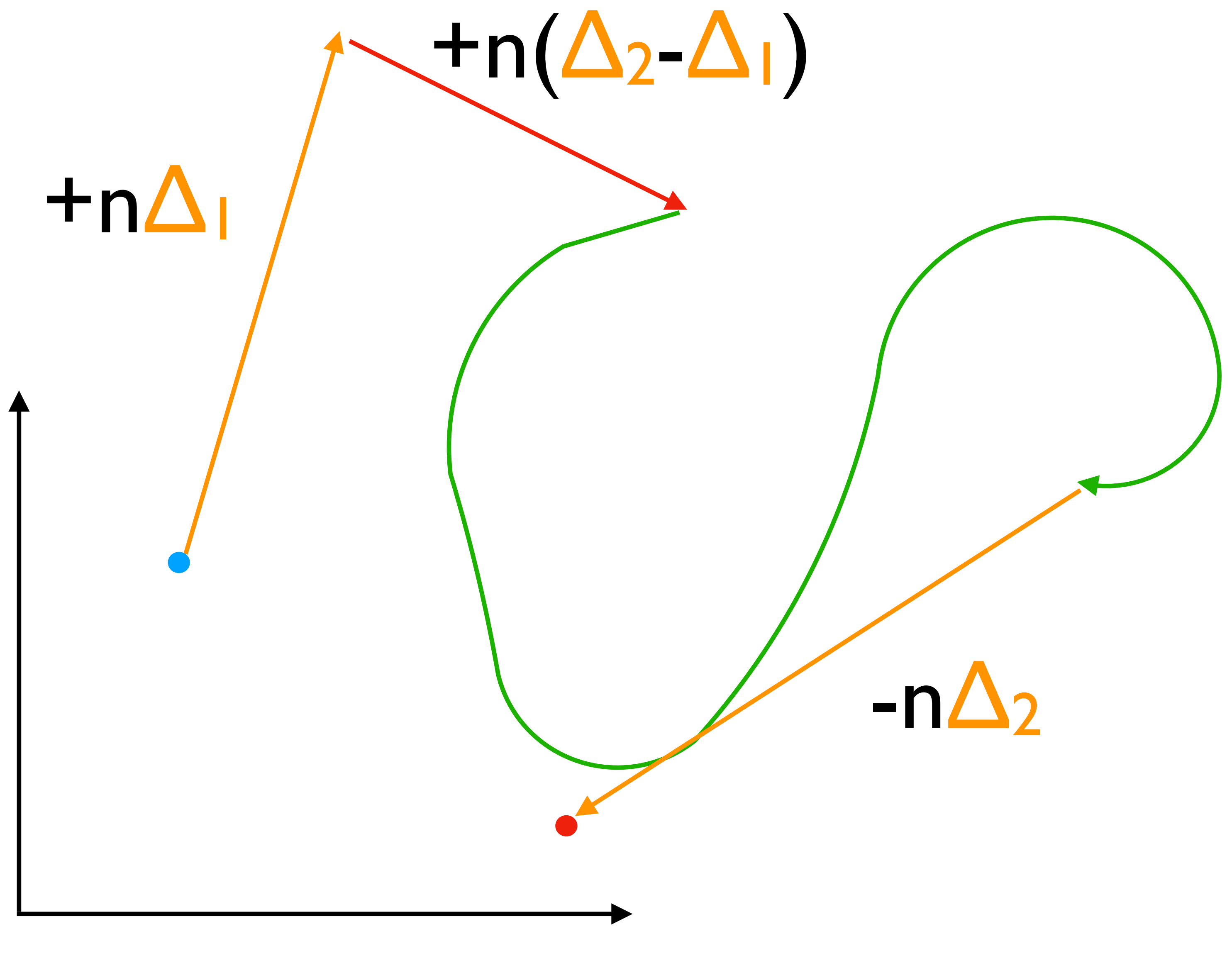
Decidability idea

$p(s) \longrightarrow p(s+\Delta_1)$ $\quad q(t+\Delta_2) \longrightarrow q(t)$ $\quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$



Decidability idea

$p(s) \longrightarrow p(s+\Delta_1)$ $\quad q(t+\Delta_2) \longrightarrow q(t)$ $\quad p(s) \xrightarrow{\mathbb{Z}} p(s+\Delta_2-\Delta_1)$



Algorithm

Algorithm

Check whether:

Algorithm

Check whether:

$$p(s) \longrightarrow p(s+\Delta_1)$$

Algorithm

Check whether:

$p(s) \longrightarrow p(s+\Delta_1)$ $\qquad$ $q(t+\Delta_2) \longrightarrow q(t)$

Algorithm

Check whether:

$p(s) \longrightarrow p(s+\Delta_1)$ $\qquad$ $q(t+\Delta_2) \longrightarrow q(t)$

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

Algorithm

Check whether:

$p(s) \longrightarrow p(s+\Delta_1)$ $\qquad$ $q(t+\Delta_2) \longrightarrow q(t)$

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

If yes then return YES

Algorithm

Check whether:

$$p(s) \longrightarrow p(s+\Delta_1) \qquad q(t+\Delta_2) \longrightarrow q(t)$$

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

If yes then return YES

If no then simplify

Algorithm

Check whether:

$$p(s) \longrightarrow p(s+\Delta_1) \qquad\qquad q(t+\Delta_2) \longrightarrow q(t)$$

$p(s) \xrightarrow{\mathbb{Z}} q(t)$ by runs using each transition many times

If yes then return YES

If no then simplify

Involved!

Ackermann-hardness

Ackermann-hardness

Theorem

The Reachability Problem for (3k+2)-VASSes is $\mathbb{F}_k$-hard.

Ackermann-hardness

Theorem

The Reachability Problem for (3k+2)-VASSes is $\mathbb{F}_k$-hard.

↑

Sławomir Lasota

Ackermann-hardness

Theorem

The Reachability Problem for (3k+2)-VASSes is $\mathbb{F}_k$-hard.

Sławomir Lasota

Cz., Łukasz Orlikowski: 6k

Ackermann-hardness

Theorem
The Reachability Problem for (3k+2)-VASSes is $\mathbb{F}_k$-hard.

↑

Sławomir Lasota

Cz., Łukasz Orlikowski: 6k

Jerome Leroux (currently): 2k+4

Big counters

Big counters

The following problem is $\mathbb{F}_k$-complete ($k \geq 3$)

Big counters

The following problem is $\mathbb{F}_k$-complete ($k \geq 3$)

Given:

Big counters

The following problem is $\mathbb{F}_k$-complete ($k \geq 3$)

Given: a two-counter automaton A with zero-tests, number n

Big counters

The following problem is $\mathbb{F}_k$-complete ($k \geq 3$)

Given: a two-counter automaton A with zero-tests, number n

Question:

Big counters

The following problem is $\mathbb{F}_k$-complete ($k \geq 3$)

Given: a two-counter automaton A with zero-tests, number n

Question: does A have an $F_k(n)$-bounded run?

Multiplication triples

Multiplication triples

Lemma

If for each n there is a d-VASS with transitions of size $\leq$ n such that

Multiplication triples

Lemma

If for each n there is a d-VASS with transitions of size ≤ n such that

arbitrary big m guessed
some counters reach 0

→

Multiplication triples

Lemma

If for each n there is a d-VASS with transitions of size ≤ n such that

arbitrary big m guessed
some counters reach 0

→ $(F_k(n), m, F_k(n)\ m)$

Multiplication triples

Lemma

If for each n there is a d-VASS with transitions of size $\leq n$ such that

arbitrary big m guessed
some counters reach 0

→ $(F_k(n), m, F_k(n)\ m)$

then reachability for d-VASSes is $\mathbb{F}_k$-hard

Multiplication triples

Lemma

If for each n there is a d-VASS with transitions of size $\leq n$ such that

arbitrary big m guessed some counters reach 0

→ $(F_k(n), m, F_k(n)\, m)$

then reachability for d-VASSes is $\mathbb{F}_k$-hard

Proof: simulate $F_k(n)$-bounded run

Triples

Triples

Triples (B, m, Bm) allow zero-testing

Triples

Triples (B, m, Bm) allow zero-testing

for m/2 zero-tests on B-bounded counters

Triples

Triples (B, m, Bm) allow zero-testing

for $m/2$ zero-tests on B-bounded counters

Goal: compute $(F_k(n), m, F_k(n)\ m)$

Triples

Triples (B, m, Bm) allow zero-testing

for $m/2$ zero-tests on B-bounded counters

Goal: compute $(F_k(n), m, F_k(n)\ m)$

For k = 1 easy: $(2n, 0, 0) + m(0, 1, 2n)$

Triples

Triples

Assume $(x, y, z) = (B, m, Bm)$

Triples

Assume $(x, y, z) = (B, m, Bm)$

Let $x' = 0$

Triples

Assume $(x, y, z) = (B, m, Bm)$

keep $x+x' = B$

Let $x' = 0$

Triples

Assume (x, y, z) = (B, m, Bm)

keep x+x’ = B

Let x’ = 0

zero-test(x’):

Triples

Assume (x, y, z) = (B, m, Bm)

keep x+x’ = B

Let x’ = 0

zero-test(x’):

loop {inc(x’), dec(x), dec(z)}

Triples

keep x+x’ = B

Assume (x, y, z) = (B, m, Bm)

Let x’ = 0

zero-test(x’):

loop {inc(x’), dec(x), dec(z)}

loop {dec(x’), inc(x), dec(z)}

Triples

Assume (x, y, z) = (B, m, Bm)

keep x+x’ = B

Let x’ = 0

zero-test(x’):

loop {inc(x’), dec(x), dec(z)}

loop {dec(x’), inc(x), dec(z)}

y := y-2

Triples

Assume (x, y, z) = (B, m, Bm)

keep x+x’ = B

Let x’ = 0

zero-test(x’):

loop {inc(x’), dec(x), dec(z)}

loop {dec(x’), inc(x), dec(z)}

y dec by 2

y := y-2

Triples

Assume (x, y, z) = (B, m, Bm)

keep x+x' = B

Let x' = 0

zero-test(x'):

loop {inc(x'), dec(x), dec(z)}

loop {dec(x'), inc(x), dec(z)}

y dec by 2

y := y-2

z dec by ≤2B

Triples

Assume (x, y, z) = (B, m, Bm)

keep x+x' = B

Let x' = 0

zero-test(x'):

loop {inc(x'), dec(x), dec(z)}

y dec by 2

loop {dec(x'), inc(x), dec(z)}

z dec by ≤2B

y := y-2

At the end check if z = 0

Testing more counters

Testing more counters

Goal: z dec by $\leq 2B$

Testing more counters

Goal: z dec by $\leq 2B$

Let $x_1 + \ldots + x_k = B$

Testing more counters

Goal: z dec by $\leq 2B$

Let $x_1 + \ldots + x_k = B$

zero-test(x_1):

Testing more counters

Goal: z dec by ≤ 2B

Let $x_1 + \ldots + x_k =$ B

zero-test(x_1):

transfer(x_2, x_1, z)

Testing more counters

Goal: z dec by $\leq 2B$

Let $x_1 + \ldots + x_k = B$

loop {dec(x_2), inc(x_1), dec(z)}

zero-test(x_1):

transfer(x_2, x_1, z)

Testing more counters

Goal: z dec by $\leq 2B$

loop {dec(x_2), inc(x_1), dec(z)}

Let $x_1 + \ldots + x_k = B$

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

Testing more counters

Goal: z dec by $\leq 2B$

loop {dec(x_2), inc(x_1), dec(z)}

Let $x_1 + \ldots + x_k = B$

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

…

Testing more counters

Goal: z dec by $\leq 2B$

Let $x_1 + \ldots + x_k = B$

loop {dec(x_2), inc(x_1), dec(z)}

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

…

transfer(x_k, x_{k-1}, z)

Testing more counters

Goal: z dec by $\leq 2B$

Let $x_1 + \ldots + x_k = B$

loop {dec(x_2), inc(x_1), dec(z)}

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

…

transfer(x_k, x_{k-1}, z)

transfer(x_{k-1}, x_k, z)

Testing more counters

Goal: z dec by ≤2B

loop {dec(x_2), inc(x_1), dec(z)}

Let $x_1 + \ldots + x_k$ = B

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

…

transfer(x_k, x_{k-1}, z)

transfer(x_{k-1}, x_k, z)

…

Testing more counters

Goal: z dec by ≤ 2B

loop {dec(x_2), inc(x_1), dec(z)}

Let $x_1 + \ldots + x_k =$ B

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

…

transfer(x_k, x_{k-1}, z)

transfer(x_{k-1}, x_k, z)

…

transfer(x_2, x_3, z)

Testing more counters

Goal: z dec by $\leq 2B$

loop {dec(x_2), inc(x_1), dec(z)}

Let $x_1 + \ldots + x_k = B$

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

...

transfer(x_k, x_{k-1}, z)

transfer(x_{k-1}, x_k, z)

...

transfer(x_2, x_3, z)

transfer(x_1, x_2, z)

Testing more counters

Goal: z dec by $\leq 2B$

loop {dec(x_2), inc(x_1), dec(z)}

Let $x_1 + \ldots + x_k = B$

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

…

transfer(x_k, x_{k-1}, z)

transfer(x_{k-1}, x_k, z)

…

transfer(x_2, x_3, z)

transfer(x_1, x_2, z)

$y := y-2$

Testing more counters

Goal: z dec by $\leq 2B$

loop {dec(x_2), inc(x_1), dec(z)}

Let $x_1 + \ldots + x_k = B$

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

...

transfer(x_k, x_{k-1}, z)

transfer(x_{k-1}, x_k, z)

...

transfer(x_2, x_3, z)

transfer(x_1, x_2, z)

y := y-2

each token moved at most twice

Testing more counters

Goal: z dec by $\leq 2B$

Let $x_1 + \ldots + x_k = B$

loop {dec(x_2), inc(x_1), dec(z)}

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

...

transfer(x_k, x_{k-1}, z)

transfer(x_{k-1}, x_k, z)

...

transfer(x_2, x_3, z)

transfer(x_1, x_2, z)

y := y-2

each token moved at most twice

z dec by $\leq 2B$

Triples

Triples

Lemma

If there is a d-VASS such that

Triples

Lemma

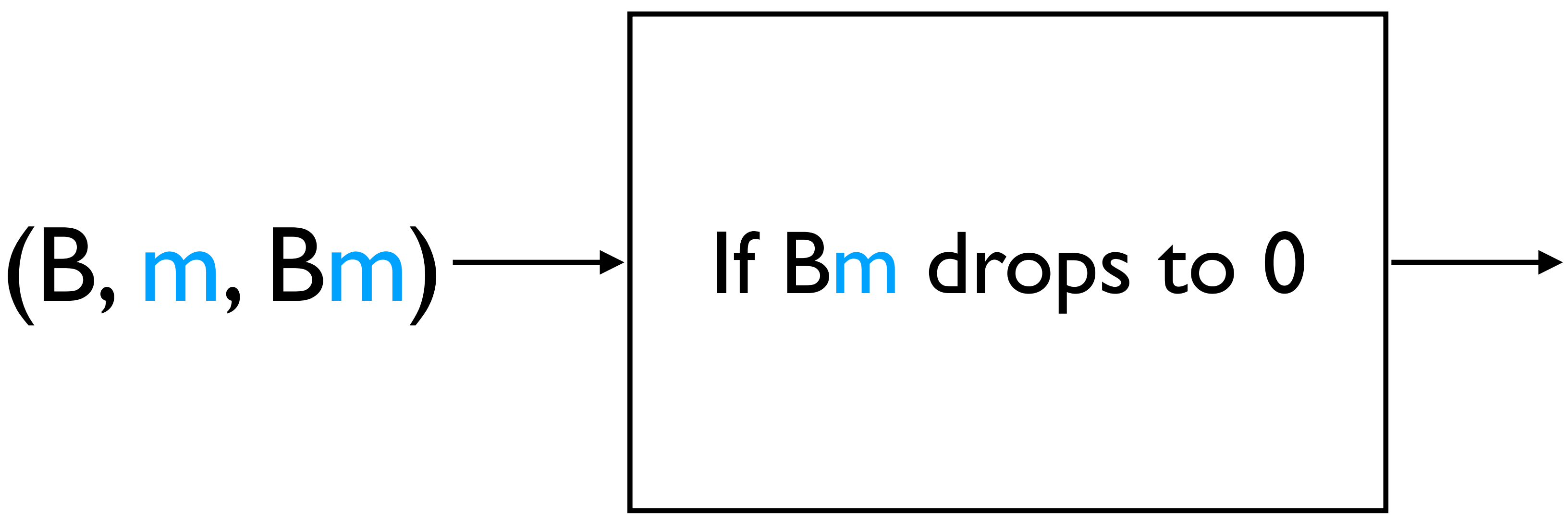
If there is a d-VASS such that

(B, m, Bm)

Triples

Lemma

If there is a d-VASS such that



Triples

Lemma

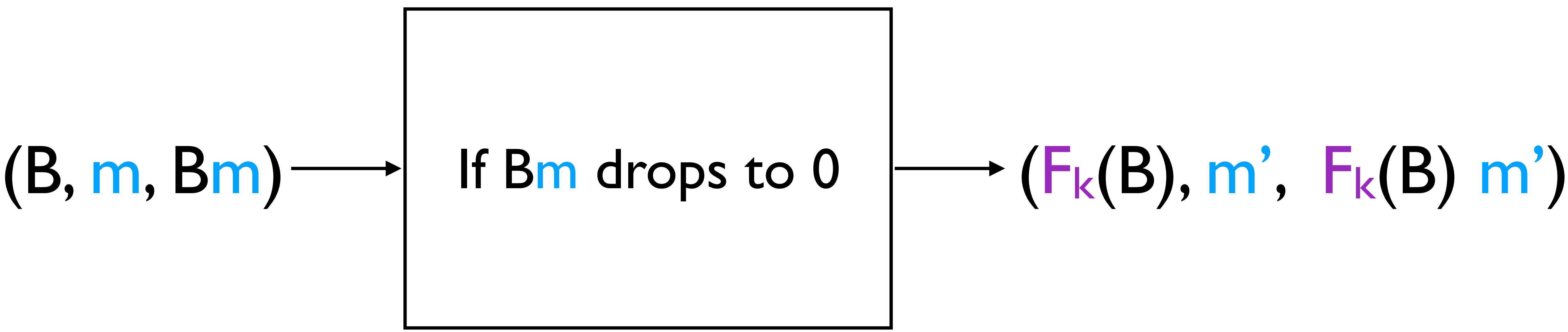
If there is a d-VASS such that

(B, m, Bm) → [If Bm drops to 0] → (F_k(B), m', F_k(B) m')

Triples

Lemma

If there is a d-VASS such that

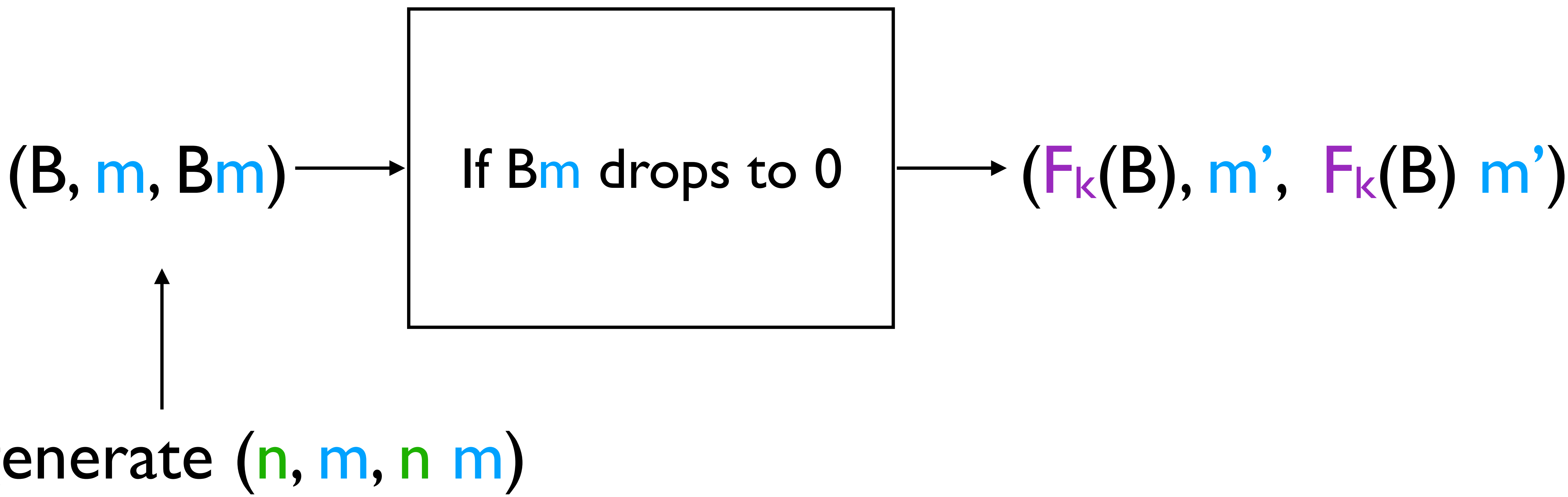


then reachability for d-VASSes is $\mathbb{F}_k$-hard.

Triples

Lemma

If there is a d-VASS such that



then reachability for d-VASSes is $\mathbb{F}_k$-hard.

Triples

Lemma

If there is a d-VASS such that

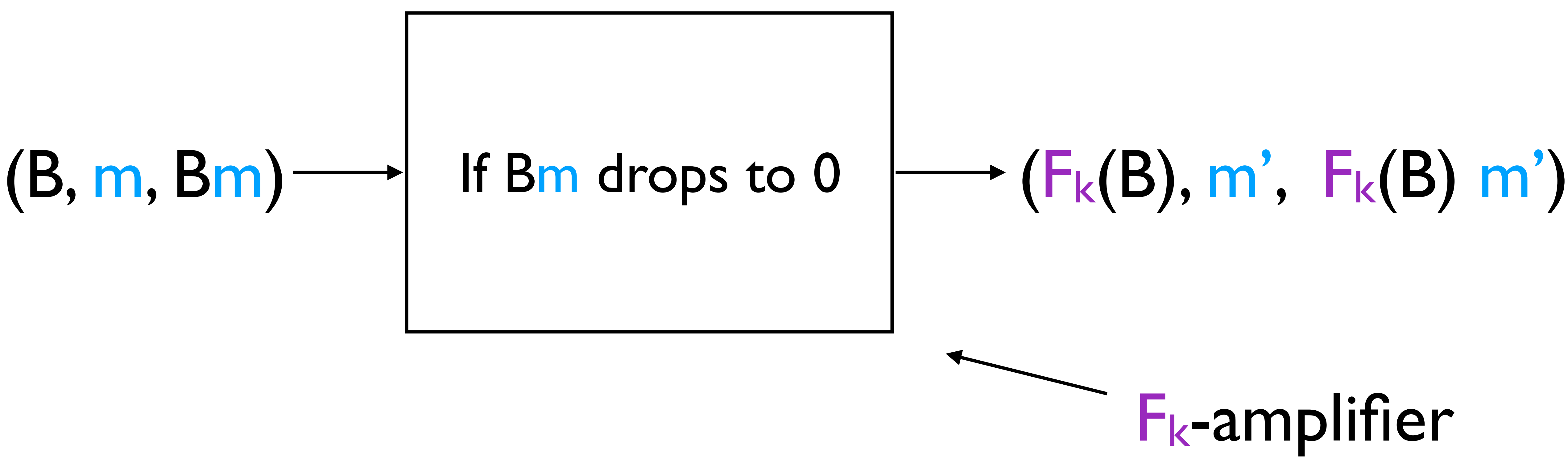
$(B, m, Bm) \rightarrow$ [If Bm drops to 0] $\rightarrow (F_k(B), m', F_k(B)\, m')$

then reachability for d-VASSes is $\mathbb{F}_k$-hard.

Triples

Lemma

If there is a d-VASS such that



then reachability for d-VASSes is $\mathbb{F}_k$-hard.

Amplifiers

Amplifiers

Lemma

F_k-amplifiers of dimension 3k+2 exist.

Amplifiers

Lemma

F_k-amplifiers of dimension $3k+2$ exist.

Proof idea: induction on k

Amplifiers

Lemma

F_k-amplifiers of dimension $3k+2$ exist.

Proof idea: induction on k

To get F_k-amplifier apply n times F_{k-1}-amplifier

General strategy

General strategy

Goal: find algorithm computing $(F_k(n), m, F_k(n)\ m)$

General strategy

Goal: find algorithm computing (F_k(n), m, F_k(n) m)

from algorithm computing (F_{k-1}(n), m, F_{k-1}(n) m)

General strategy

Goal: find algorithm computing $(F_k(n), m, F_k(n)\ m)$

from algorithm computing $(F_{k-1}(n), m, F_{k-1}(n)\ m)$

$$F_k(n) = F_{k-1} \circ \ldots \circ F_{k-1}(1)$$

General strategy

Goal: find algorithm computing $(F_k(n), m, F_k(n)\ m)$

from algorithm computing $(F_{k-1}(n), m, F_{k-1}(n)\ m)$

$$F_k(n) = F_{k-1} \circ \ldots \circ F_{k-1}(1)$$

Idea:

General strategy

Goal: find algorithm computing $(F_k(n), m, F_k(n)\ m)$

from algorithm computing $(F_{k-1}(n), m, F_{k-1}(n)\ m)$

$$F_k(n) = F_{k-1} \circ \ldots \circ F_{k-1}(1)$$

Idea: start from triple $(1, m, m)$

General strategy

Goal: find algorithm computing $(F_k(n), m, F_k(n)\ m)$

from algorithm computing $(F_{k-1}(n), m, F_{k-1}(n)\ m)$

$$F_k(n) = F_{k-1} \circ \ldots \circ F_{k-1}(1)$$

Idea: start from triple $(1, m, m)$

n times apply F_{k-1} to 1

F_k-amplifiers

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m')

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m') guess (1, M, M)

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m')   guess (1, M, M)

repeat B/12 times:

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m') guess (1, M, M)

repeat B/12 times:

apply (x, y, z) $\xrightarrow{F_{k-1}\text{-amplifier}}$ (x', y', z')

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m') guess (1, M, M)

repeat B/12 times:

apply (x, y, z) —F_{k-1}-amplifier→ (x', y', z')

zero-test x, y, z

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m') guess (1, M, M)

repeat B/12 times:

apply (x, y, z) $\xrightarrow{F_{k-1}\text{-amplifier}}$ (x', y', z')

zero-test x, y, z

transfer back x' to x, y' to y, z' to z

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m') guess (1, M, M)

repeat B/12 times:

apply (x, y, z) $\xrightarrow{F_{k-1}\text{-amplifier}}$ (x', y', z')

zero-test x, y, z

transfer back x' to x, y' to y, z' to z

zero-test x', y', z'

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m') guess (1, M, M)

repeat B/12 times:

apply (x, y, z) —F_{k-1}-amplifier→ (x', y', z')

zero-test x, y, z

transfer back x' to x, y' to y, z' to z

zero-test x', y', z'

we have (x, y, z) = (F_k(B), m', F_k(B) m')

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m') guess (1, M, M)

repeat B/12 times:

apply (x, y, z) —F_{k-1}-amplifier→ (x', y', z')

zero-test x, y, z

transfer back x' to x, y' to y, z' to z

zero-test x', y', z'

F'_k-amplifiers

we have (x, y, z) = (F_k(B), m', F_k(B) m')

F_k-amplifiers

(B, m, Bm) gives B/2 zero-tests on m-bounded counters

goal: (F_k(B), m', F_k(B) m')   guess (1, M, M)

repeat B/12 times:

apply (x, y, z) —F_{k-1}-amplifier→ (x', y', z')

zero-test x, y, z

transfer back x' to x, y' to y, z' to z

zero-test x', y', z'

we have (x, y, z) = (F_k(B), m', F_k(B) m')

Open problems

Open problems

- reachability for 3-VASSes (Tower? PSpace?)

Open problems

- reachability for 3-VASSes (Tower? PSpace?)
- $\mathbb{F}_d$-hardness in dimension d+C (in $\mathbb{F}_d$ for d-4)?

Open problems

- reachability for 3-VASSes (Tower? PSpace?)
- $\mathbb{F}_d$-hardness in dimension d+C (in $\mathbb{F}_d$ for d-4)?
- reachability for pushdown VASSes (decidable?)

Open problems

- reachability for 3-VASSes (Tower? PSpace?)
- $\mathbb{F}_d$-hardness in dimension d+C (in $\mathbb{F}_d$ for d-4)?
- reachability for pushdown VASSes (decidable?)
- hyper-Ackermann-hardness for pushdown VASSes?

Open problems

- reachability for 3-VASSes (Tower? PSpace?)
- $\mathbb{F}_d$-hardness in dimension d+C (in $\mathbb{F}_d$ for d-4)?
- reachability for pushdown VASSes (decidable?)
- hyper-Ackermann-hardness for pushdown VASSes?
- low dimensions

Open problems

- reachability for 3-VASSes (Tower? PSpace?)
- $\mathbb{F}_d$-hardness in dimension d+C (in $\mathbb{F}_d$ for d-4)?
- reachability for pushdown VASSes (decidable?)
- hyper-Ackermann-hardness for pushdown VASSes?
- low dimensions

Thank you!