

History- Deterministic One- Counter Nets

Aditya Prakash, K.S. Thejaswini

University of Warwick, UK

Deterministic Models

Algorithmically efficient,
better closure properties

Non-deterministic Models

Succinct, Expressive

Deterministic Models

Algorithmically efficient,
better closure properties

History-determinism

Non-deterministic Models

Succinct, Expressive

History-Deterministic One-Counter Nets

- What is history-determinism?
- The resolvers in an HD-OCN are semilinear
Corollary: Every HD-OCN can be converted to
a deterministic one-counter automaton.
- Complexity of checking history-determinism.

1. History - Determinism

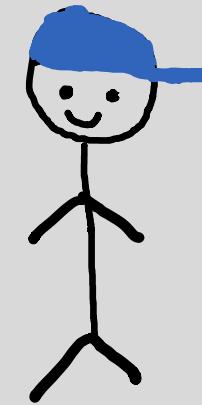
System



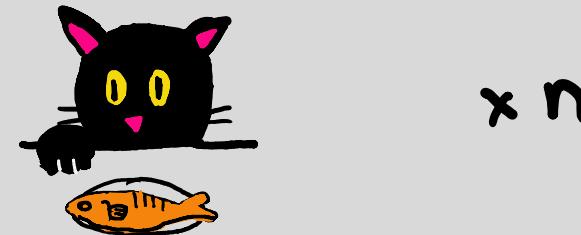
Instruction
←

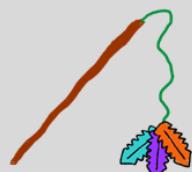
Action
→

Environment



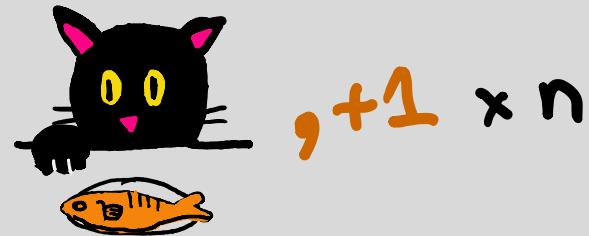
1.  $\times n$

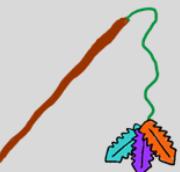


2.  $\times m$



1.  $\times n$

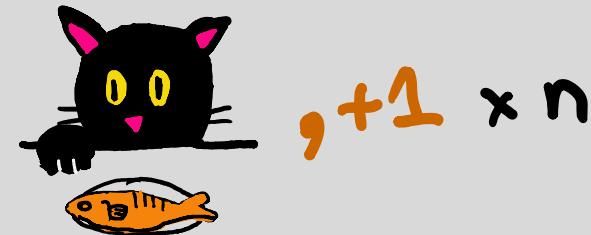


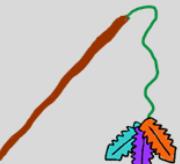
2.  $\times m$

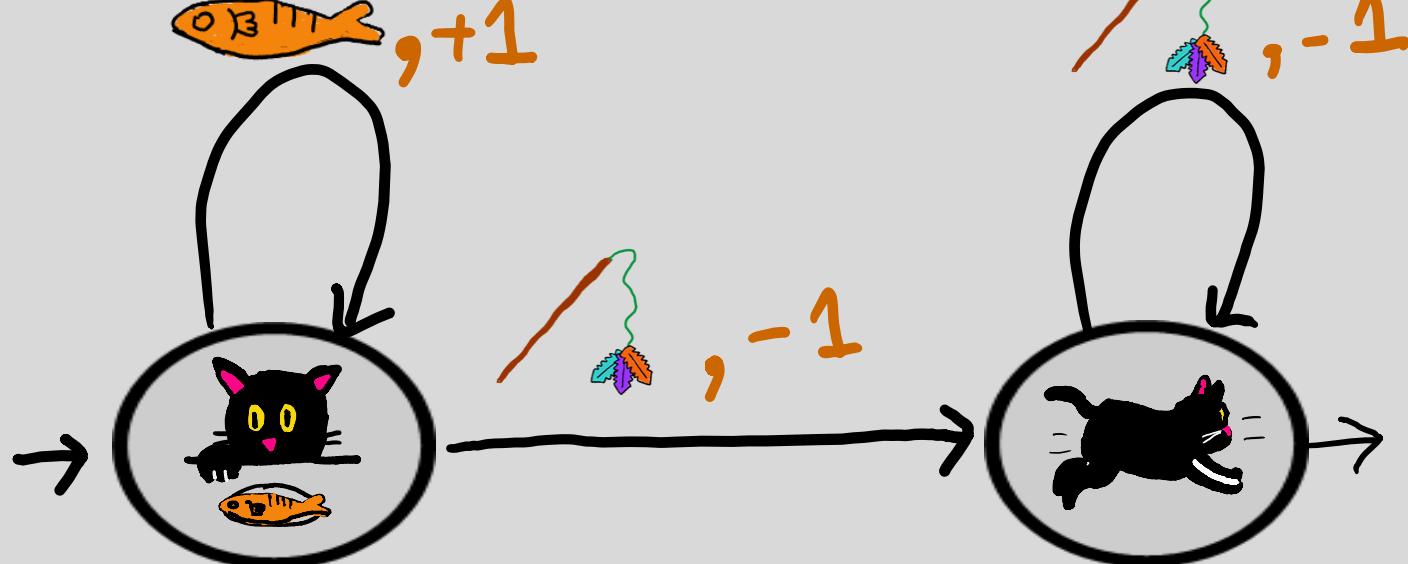


$n \geq m \geq 1$

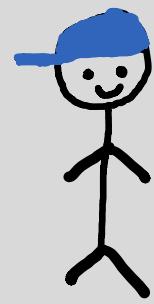
1.  $\times n$



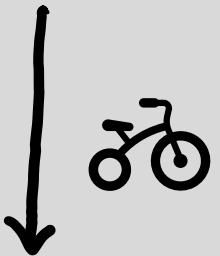
2.  $\times m$



One-Counter
Net

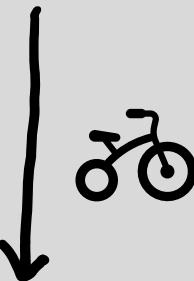


1.



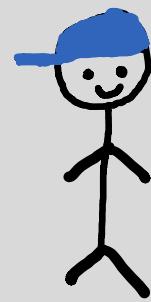
2.

Park 1

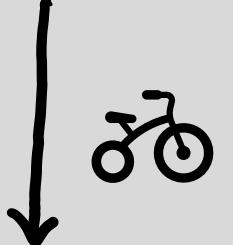


3.

Park 2

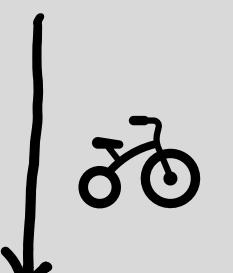


1.

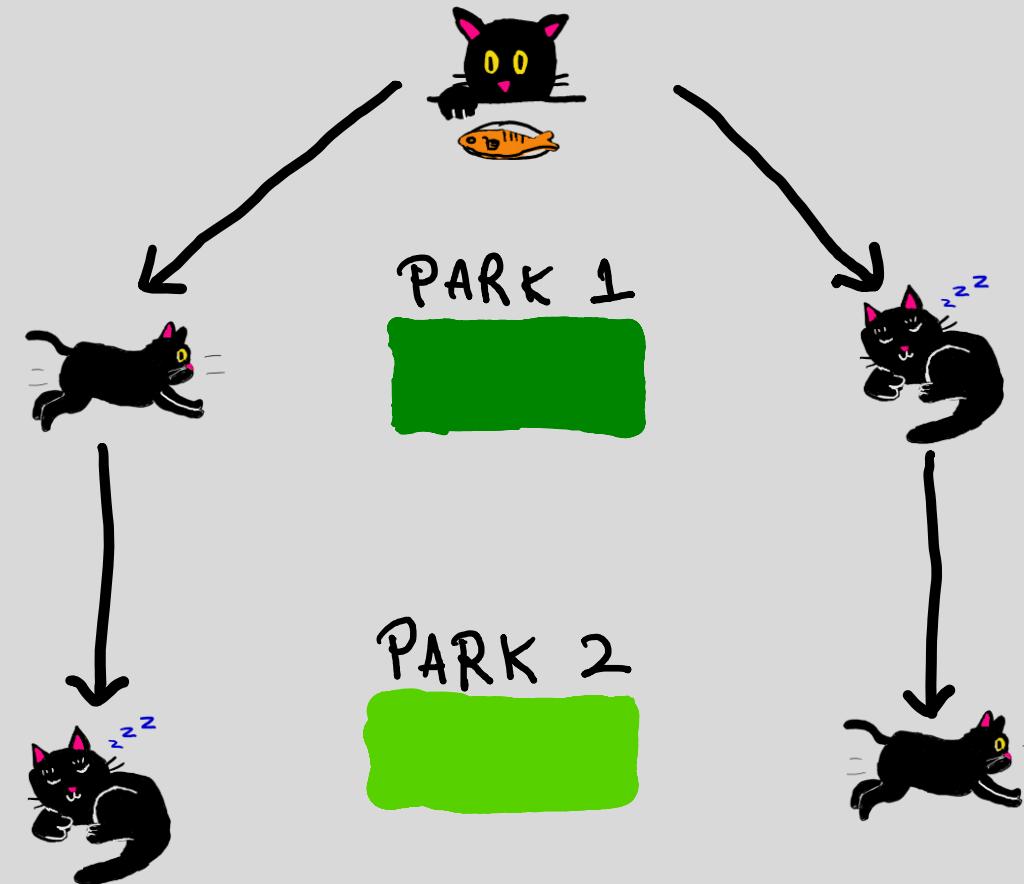


Park 1

2.

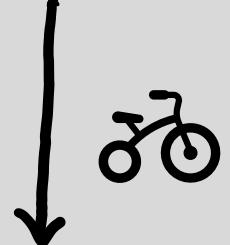


Park 2



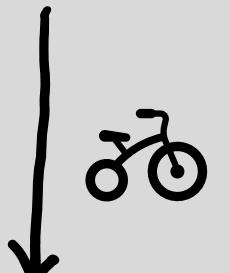


1.



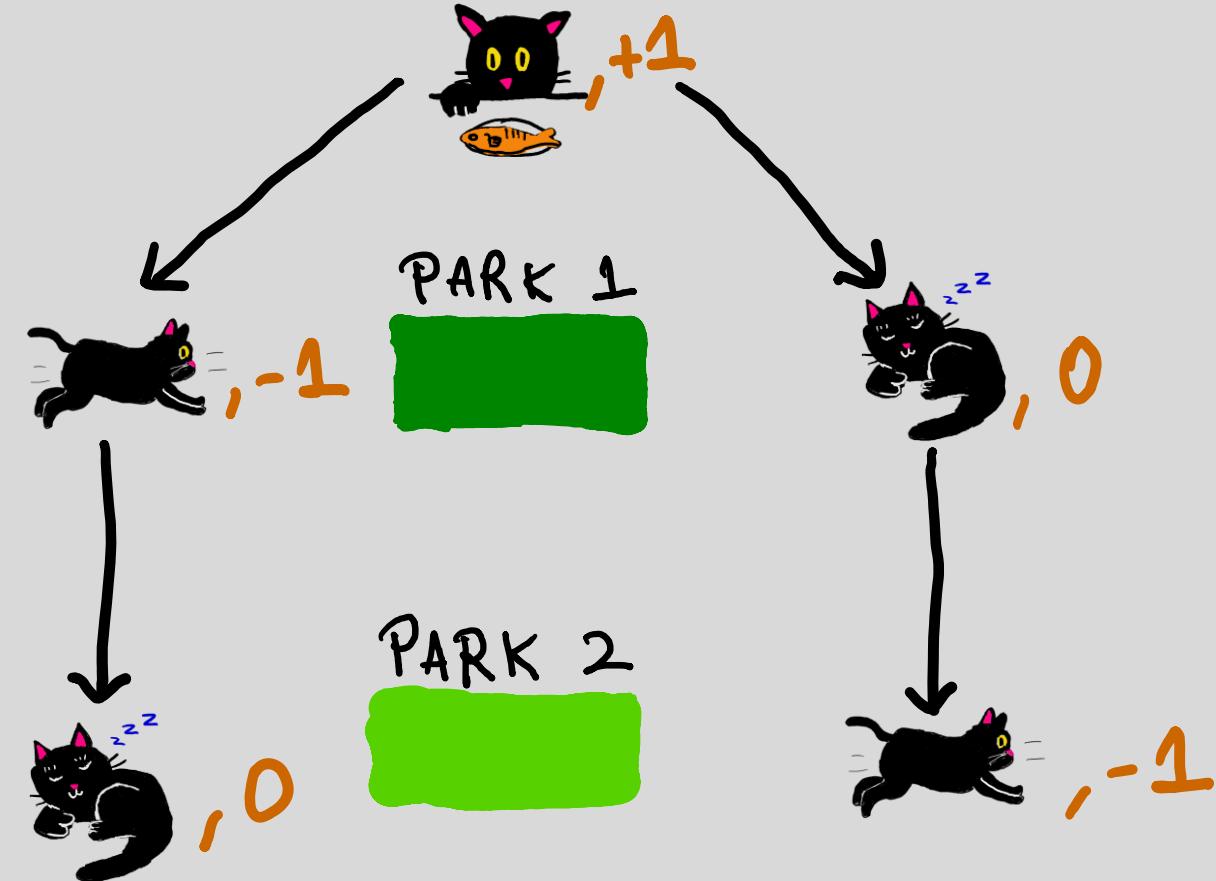
2.

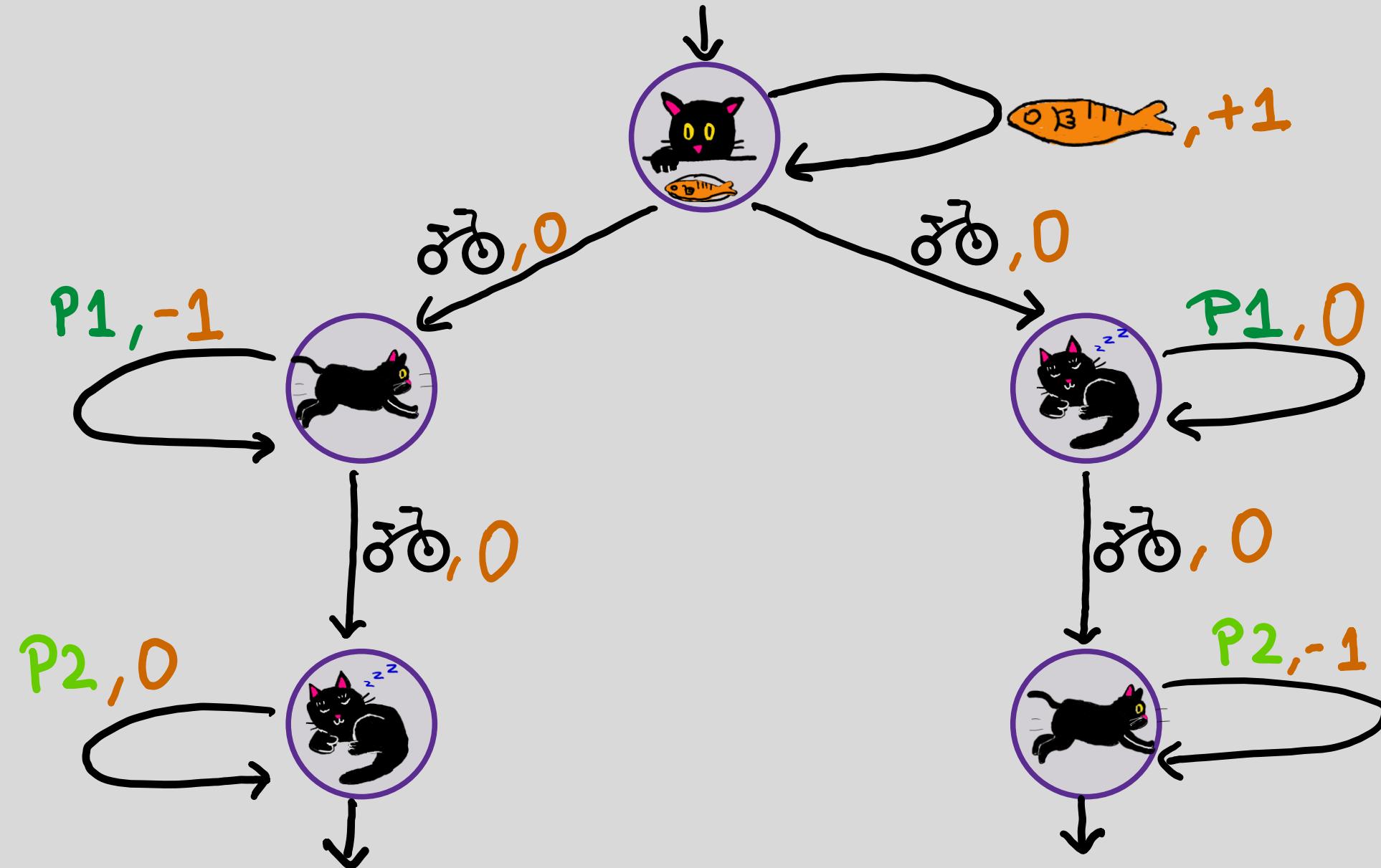
Park 1

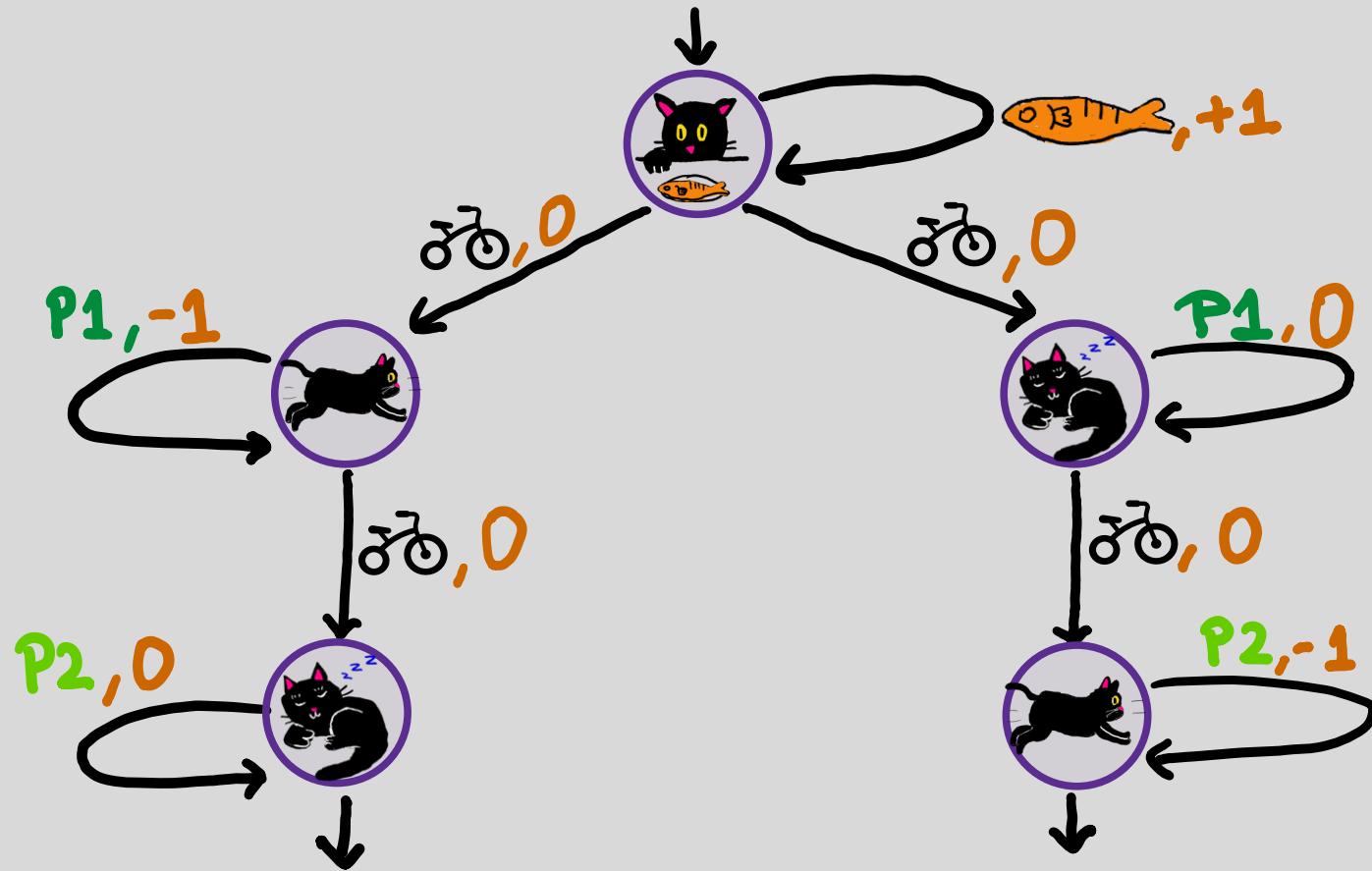


3.

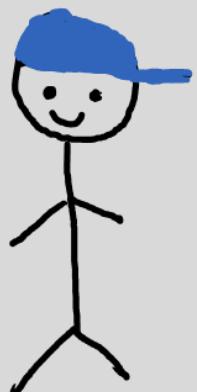
Park 2





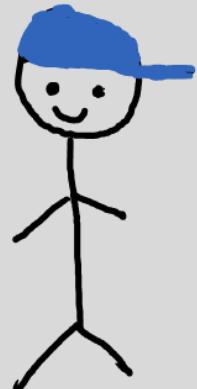


$$L = \left\{ \text{fish}^i \circledast P1^j \circledast P2^k \mid i \geq j \text{ or } i \geq k \right\}$$



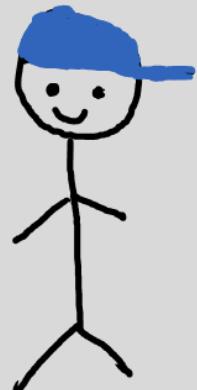
(with 5)

I will spend 4 hours
in Park1, 7 hours
in Park2.



(with 5)

I will spend 4 hours
in Park 1, 7 hours
in Park 2.

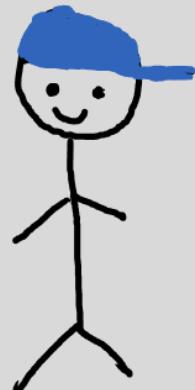


OK, I will
play in Park 1.



(with 5)

I will spend 4 hours
in Park 1, 7 hours
in Park 2.

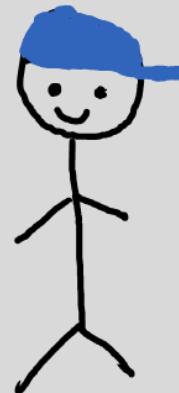


OK, I will
play in Park 1.



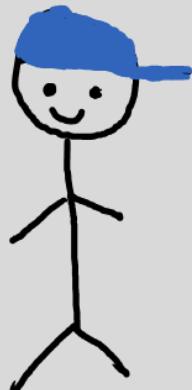
(with 5)

I will spend 4 hours
in Park 1, 3 hours
in Park 2.



(with 5)

I will spend 4 hours
in Park 1, 7 hours
in Park 2.

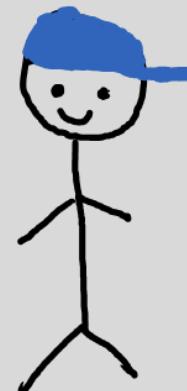


OK, I will
play in Park 1.



(with 5)

I will spend 4 hours
in Park 1, 3 hours
in Park 2.

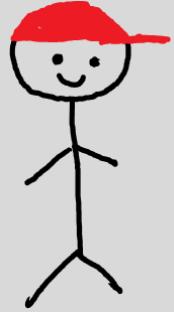


I can play
in either parks!!

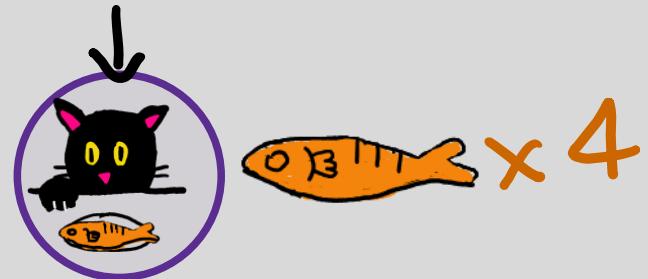
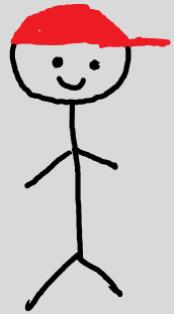


(with 5)

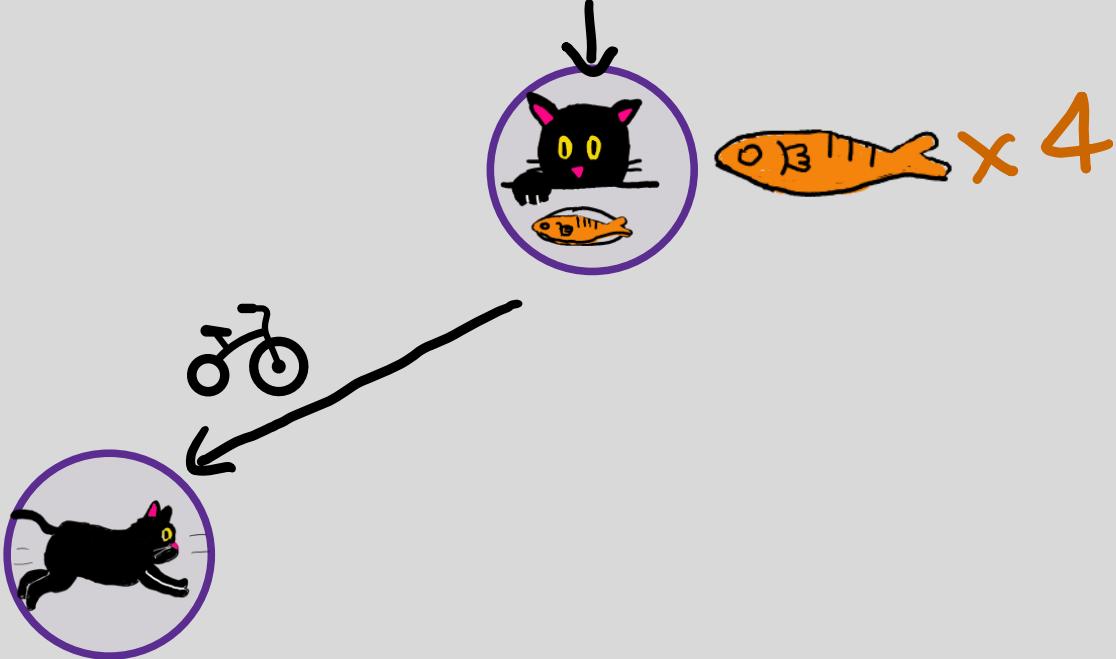
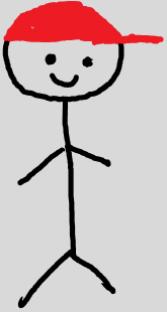
Tec



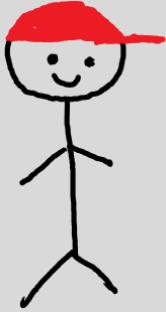
Tec



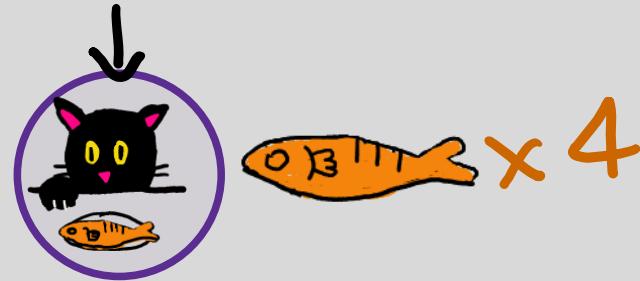
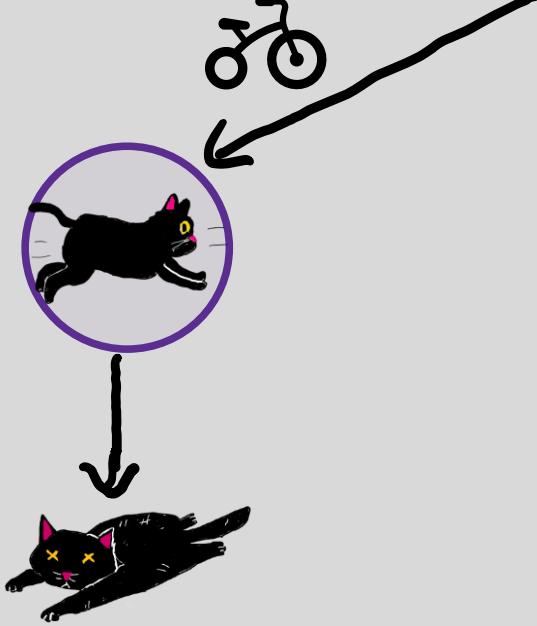
Tec



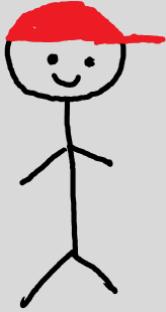
Tec



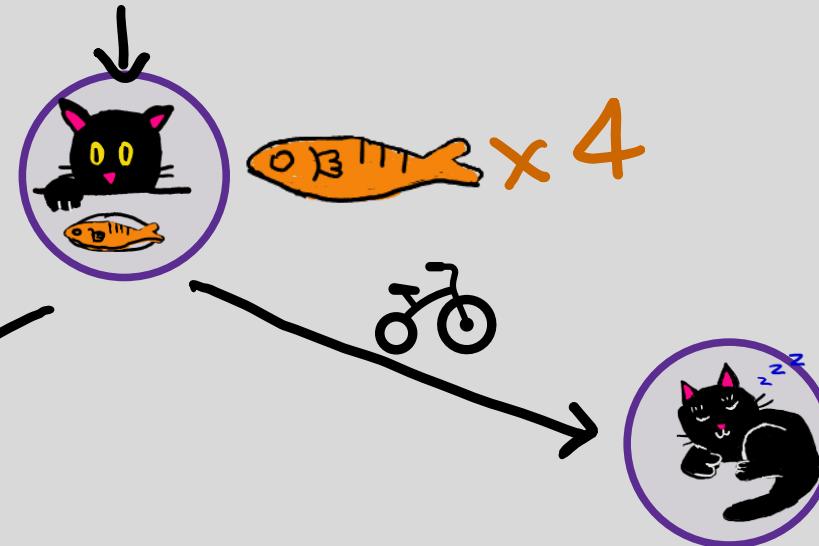
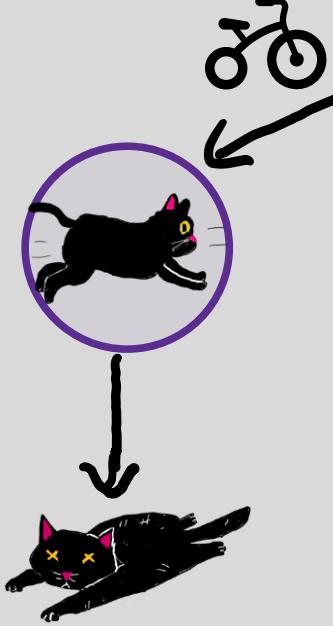
P1 x 5



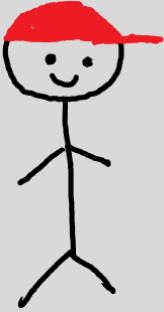
Tec



P1x5



Tec



P1 × 5



0 3 x 4



P1 × 3



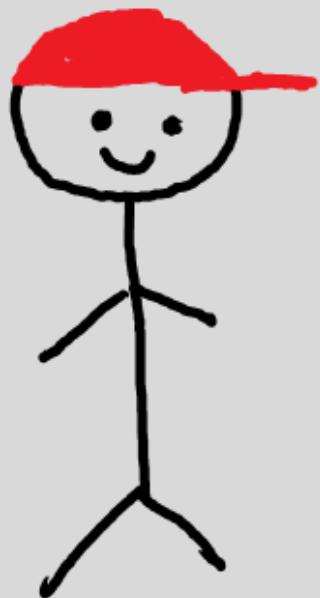
P2 × 7



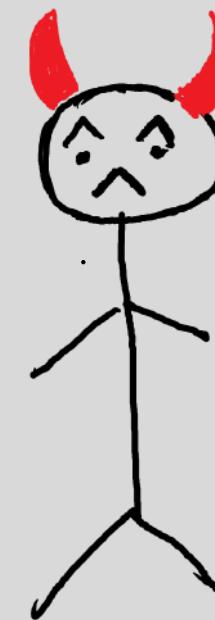
Tec



Tec



The Evil
Cat sitter



Definition: We say an one-counter net is history-deterministic if  has a strategy that produces an accepting run whenever  gives an accepting word.

History - Determinism Game:

History - Determinism Game:

Starts at (, 0):

1.  selects letter
2.  selects transition

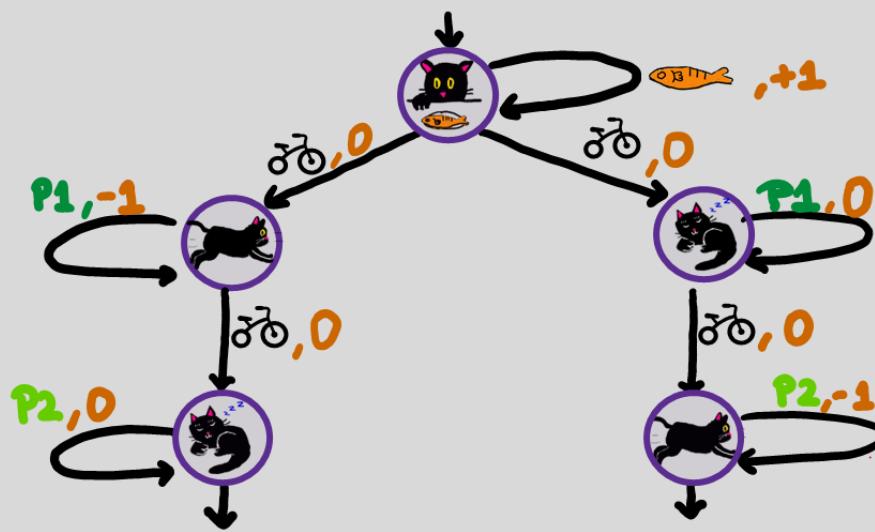
History - Determinism Game:

Starts at (, 0):

1.  selects letter

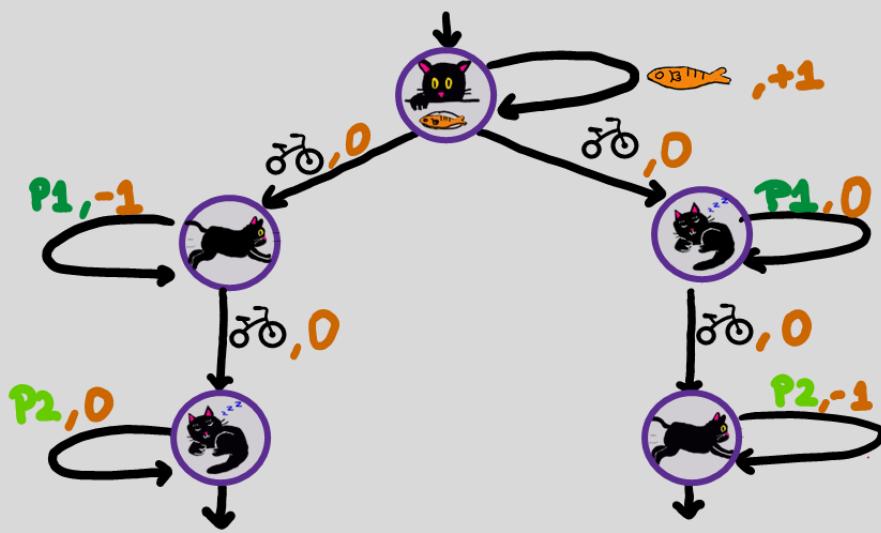
2.  selects transition

Winning Condition for : If 's word is accepting,
and 's run is rejecting.

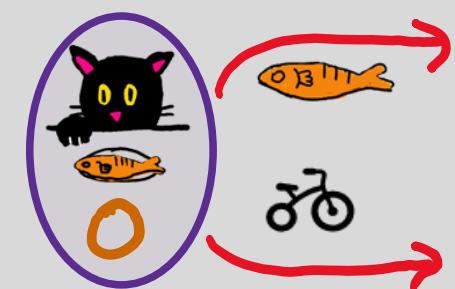


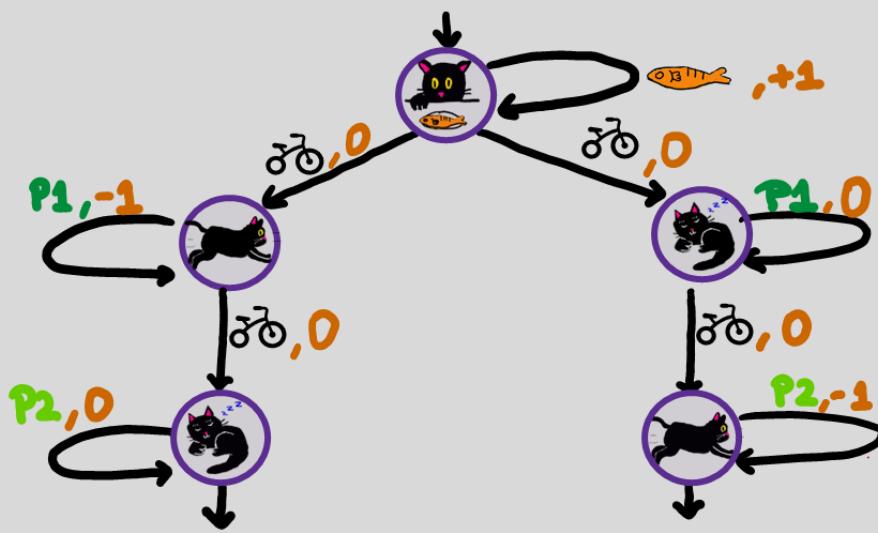
$$L = \left\{ \begin{array}{l} \text{fish } P1^i \text{ fish } P2^k | i \geq j \text{ or } i \geq k \end{array} \right\}$$



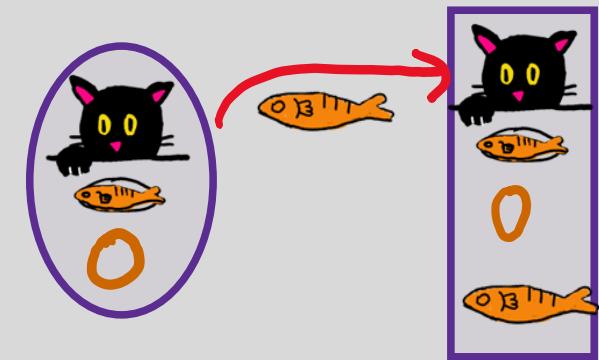


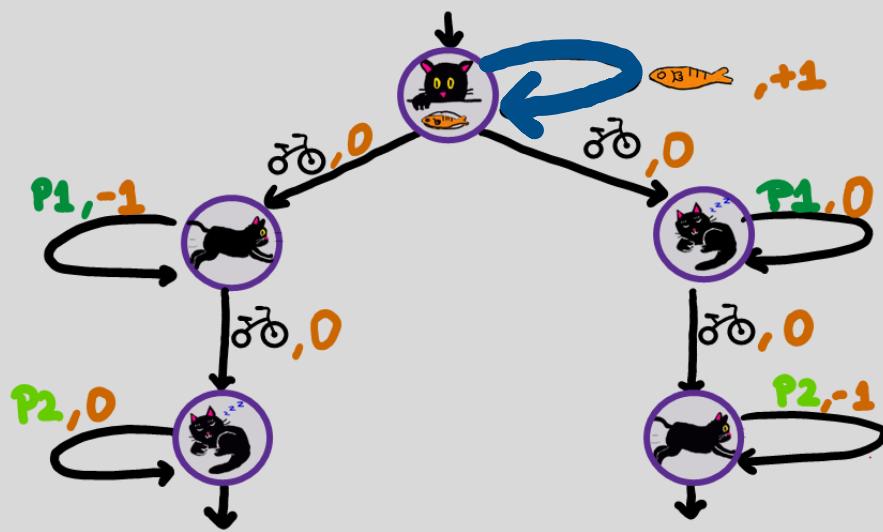
$$L = \left\{ \begin{matrix} i \\ \text{fish} \end{matrix} \middle| P_1^j \text{ and } P_2^k \mid i \geq j \text{ or } i \geq k \right\}$$



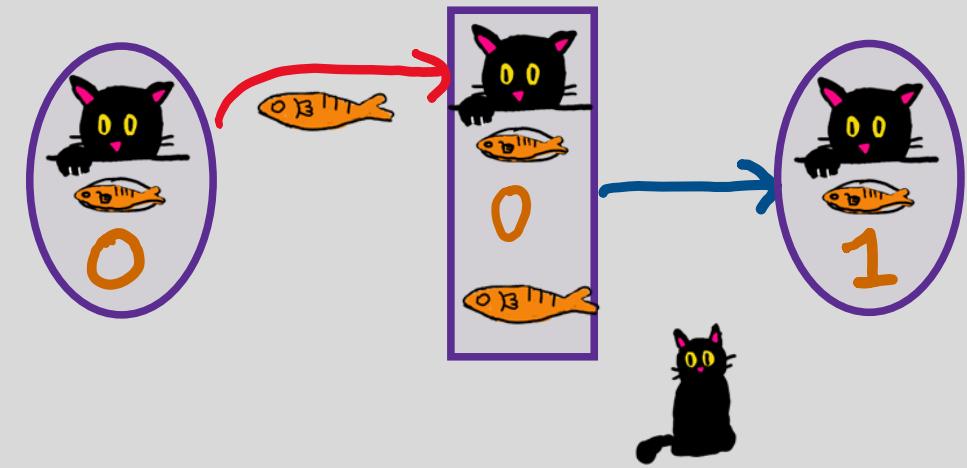


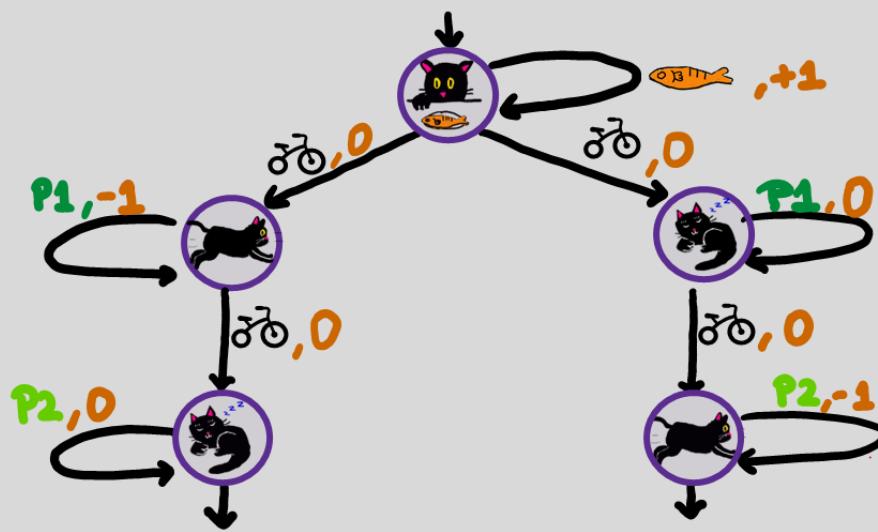
$$L = \left\{ \begin{matrix} i \\ \text{fish} \end{matrix} \middle| P_1^j \text{ and } P_2^k \mid i \geq j \text{ or } i \geq k \right\}$$



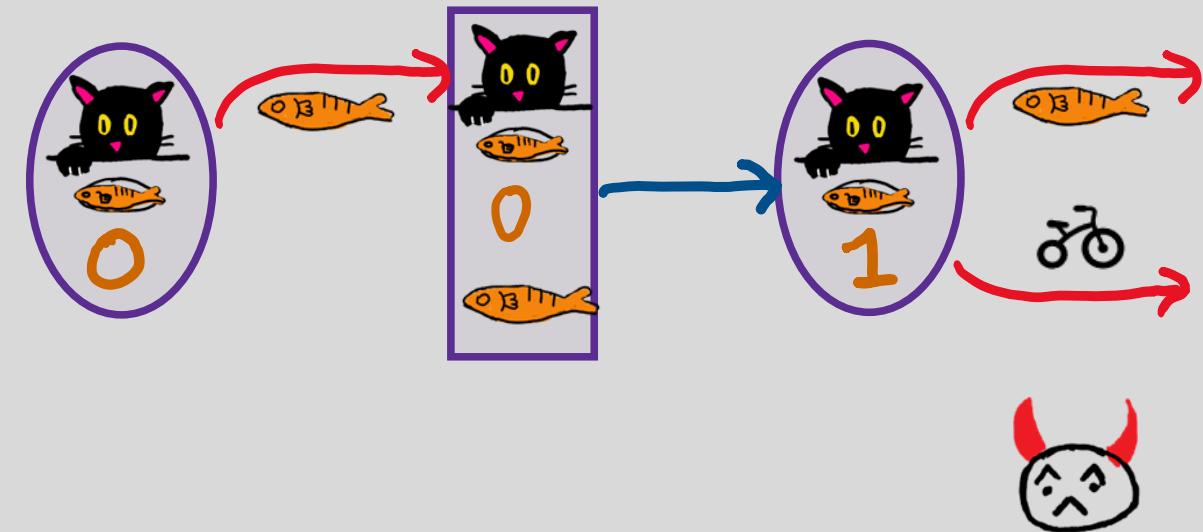


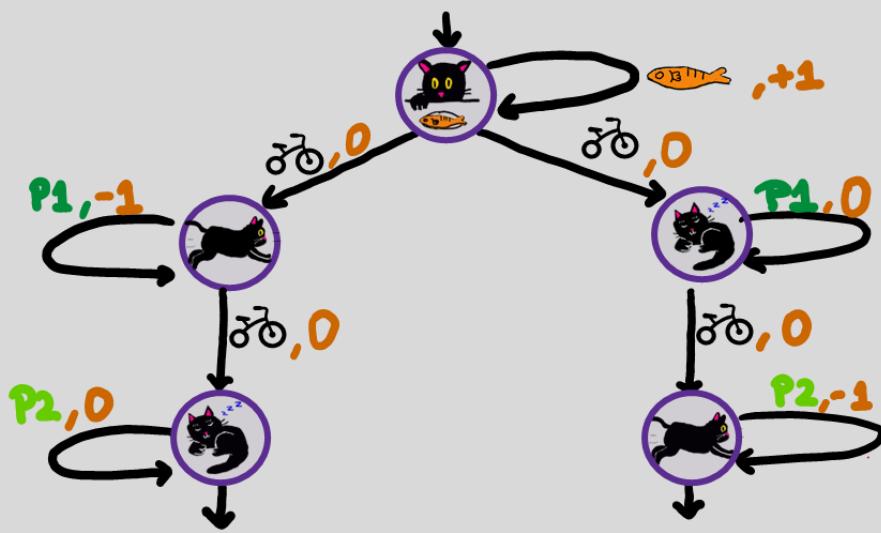
$$L = \left\{ \text{fish}^i \circledast P_1^j \circledast P_2^k \mid i \geq j \text{ or } i \geq k \right\}$$



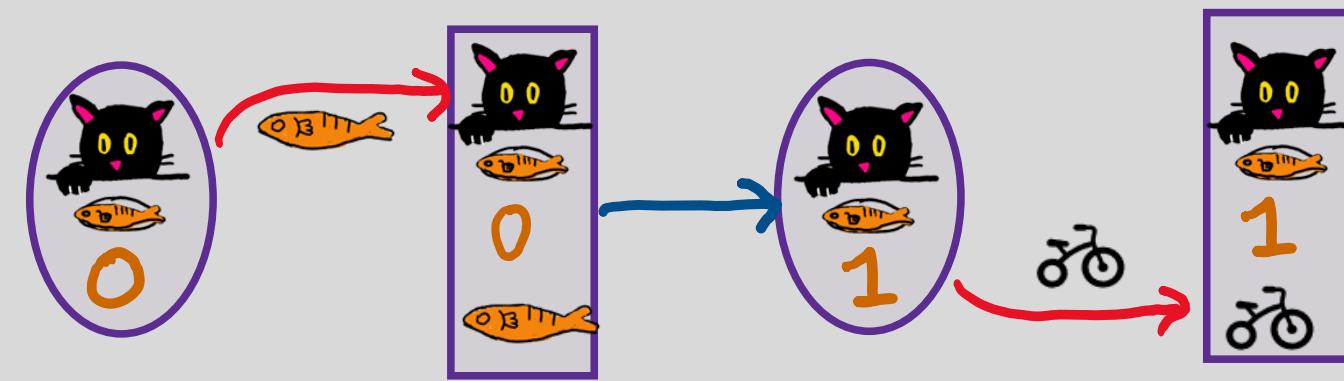


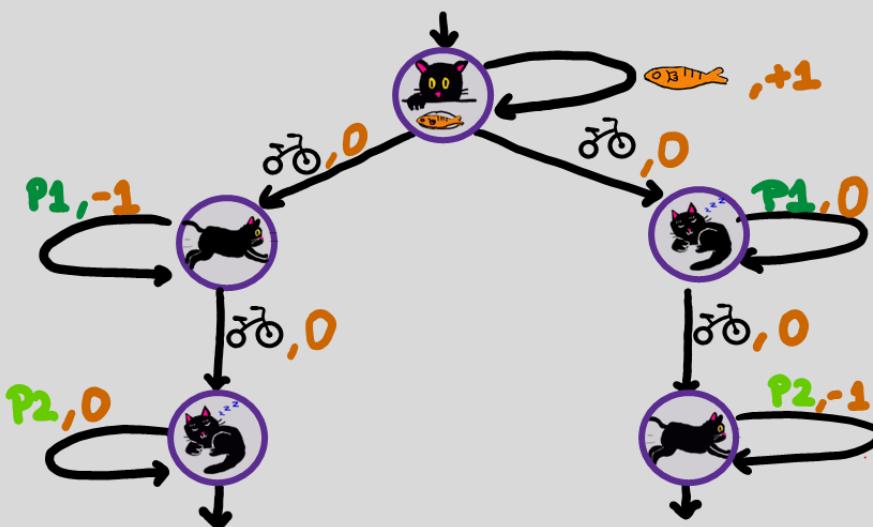
$$L = \left\{ \begin{array}{l} i \\ \text{or } P_1^j \text{ or } P_2^k \end{array} \mid i \geq j \text{ or } i \geq k \right\}$$



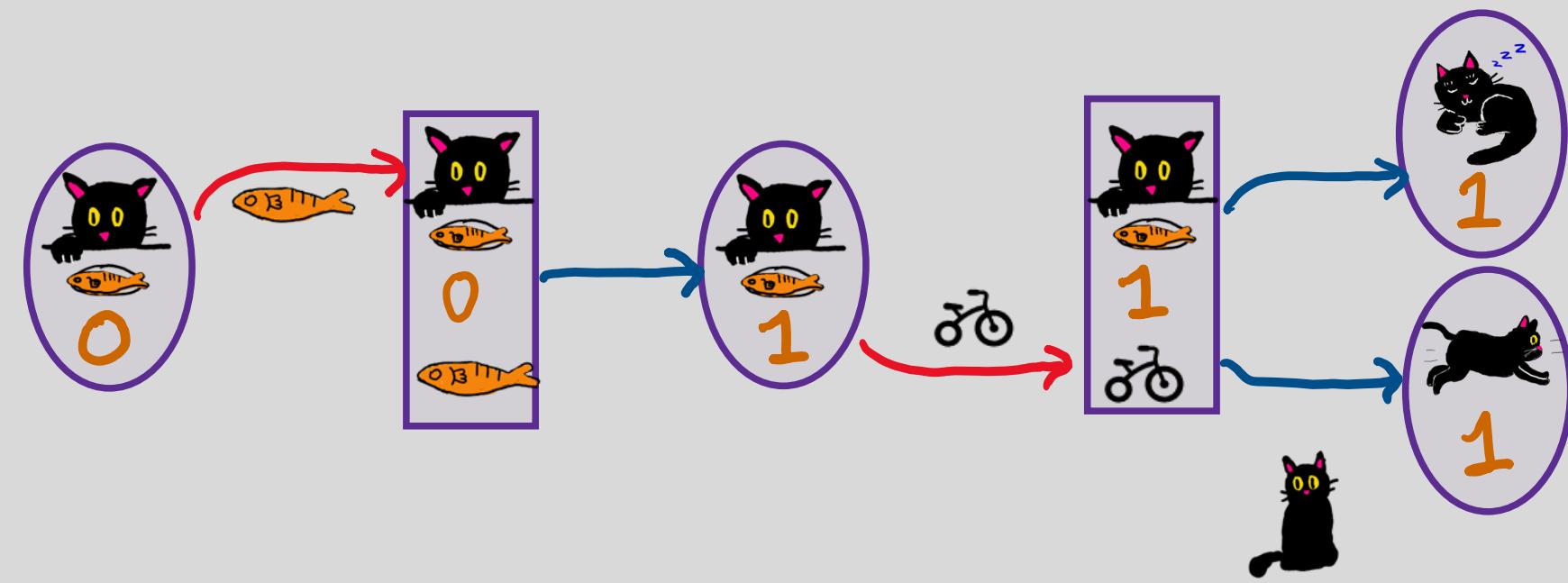


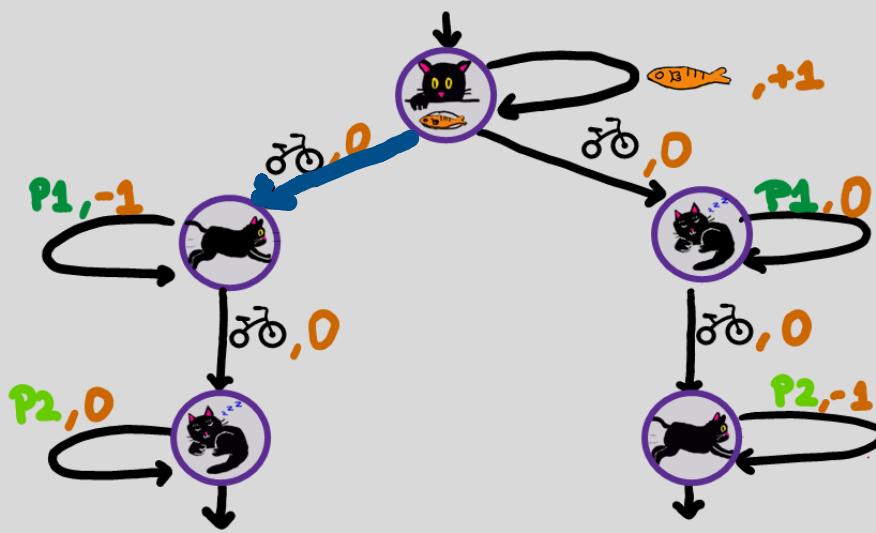
$$L = \left\{ \begin{matrix} i \\ \text{fish} \end{matrix} \middle| P_1^j \text{ and } P_2^k \mid i \geq j \text{ or } i \geq k \right\}$$



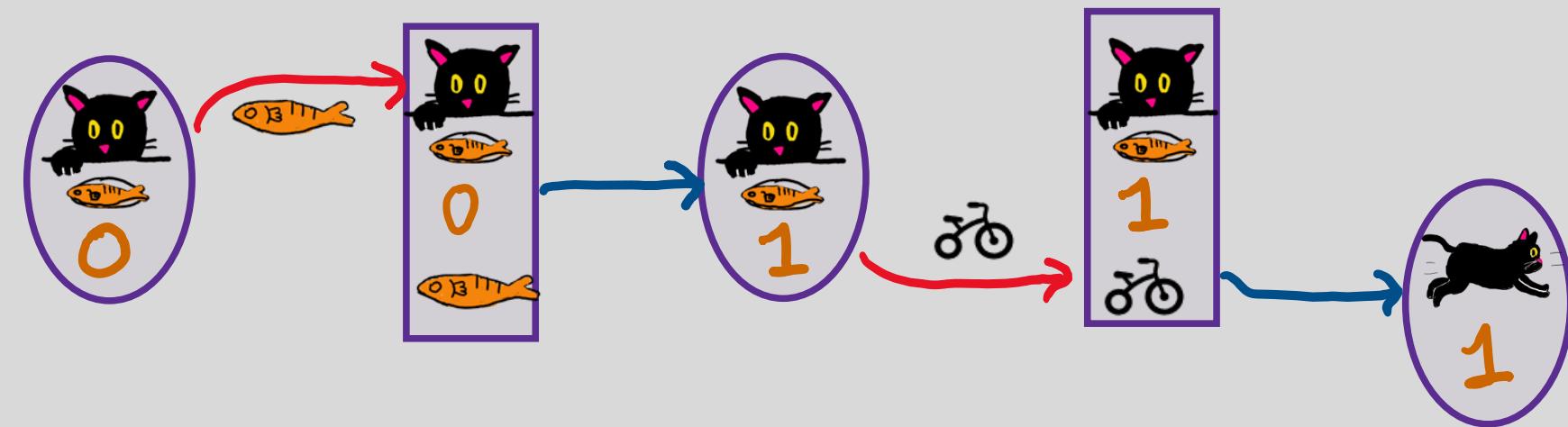


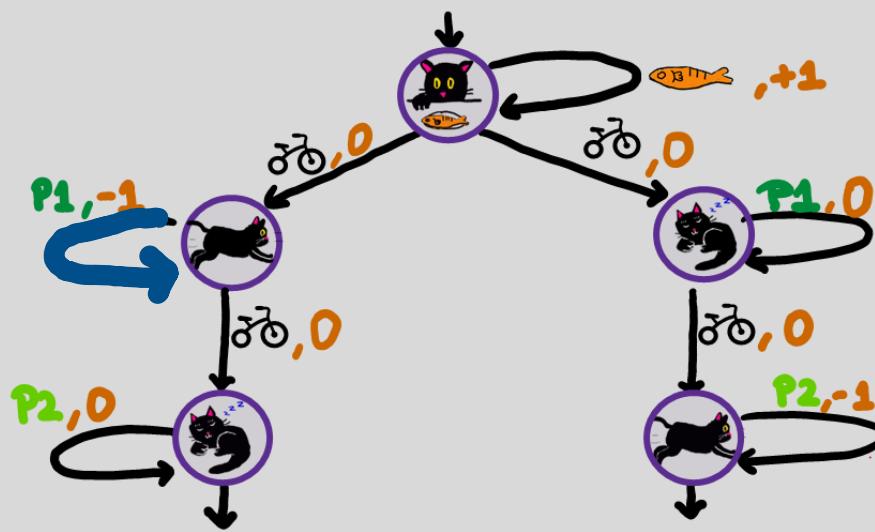
$$L = \left\{ \text{fish}^i \text{ } \circledast P_1^j \text{ } \circledast P_2^k \mid i \geq j \text{ or } i \geq k \right\}$$



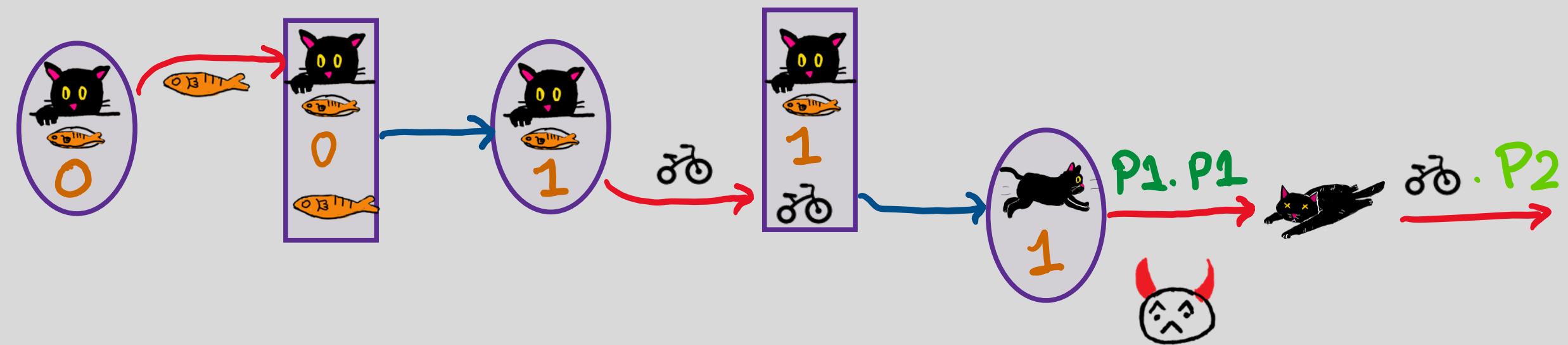


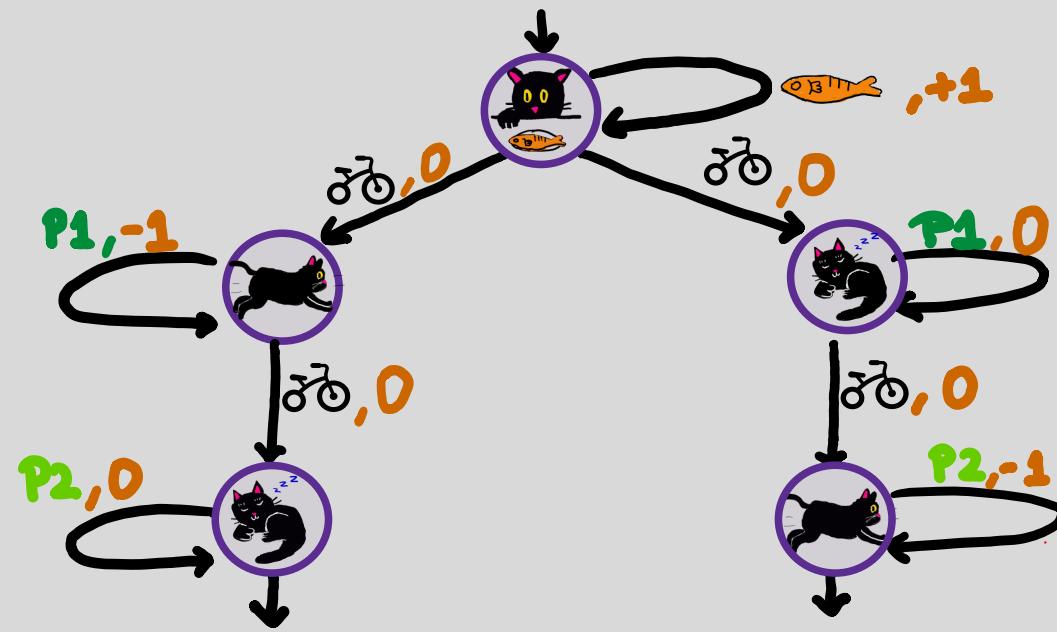
$$L = \left\{ \text{fish}^i \text{ } \text{bicycle}^j \text{ } \text{fish}^k \mid i \geq j \text{ or } i \geq k \right\}$$



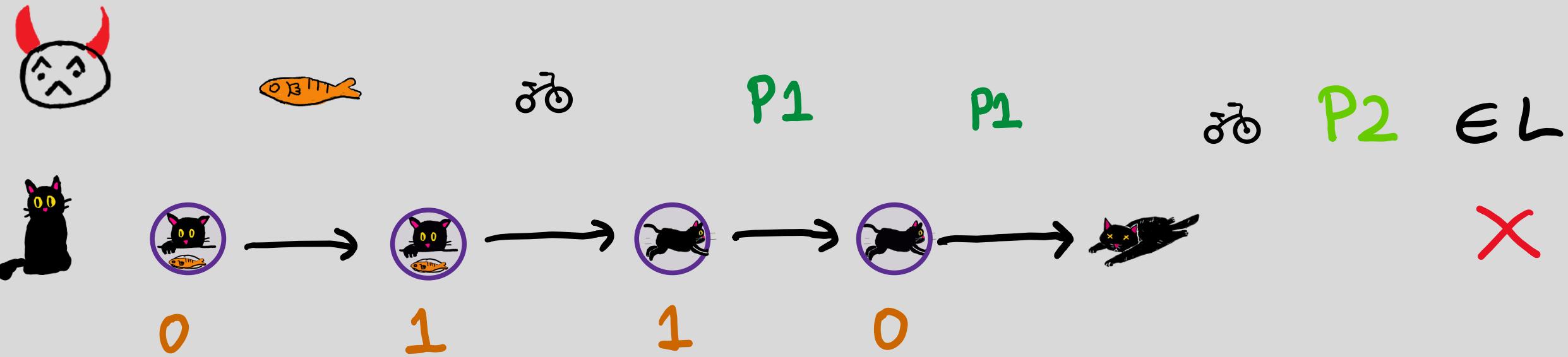


$$L = \left\{ \begin{matrix} i \\ \text{fish} \end{matrix} \middle| P_1^j \text{ fish } P_2^k \mid i \geq j \text{ or } i \geq k \right\}$$





$$L = \{ \underbrace{\text{Fish}}_i \xrightarrow{\text{Bike}} \underbrace{P1}_j \xrightarrow{\text{Bike}} \underbrace{P2}_k \mid i \geq j \text{ or } i \geq k \}$$



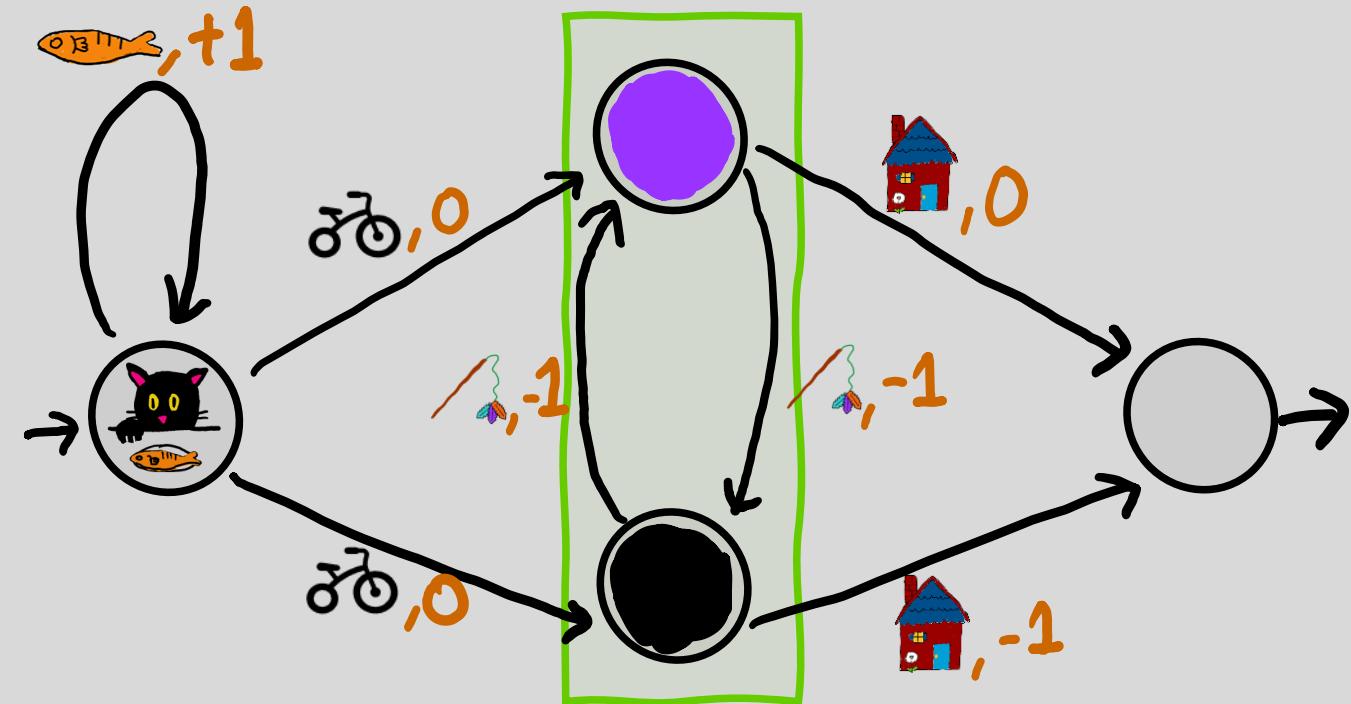
Definition: A One-counter net N is HD iff

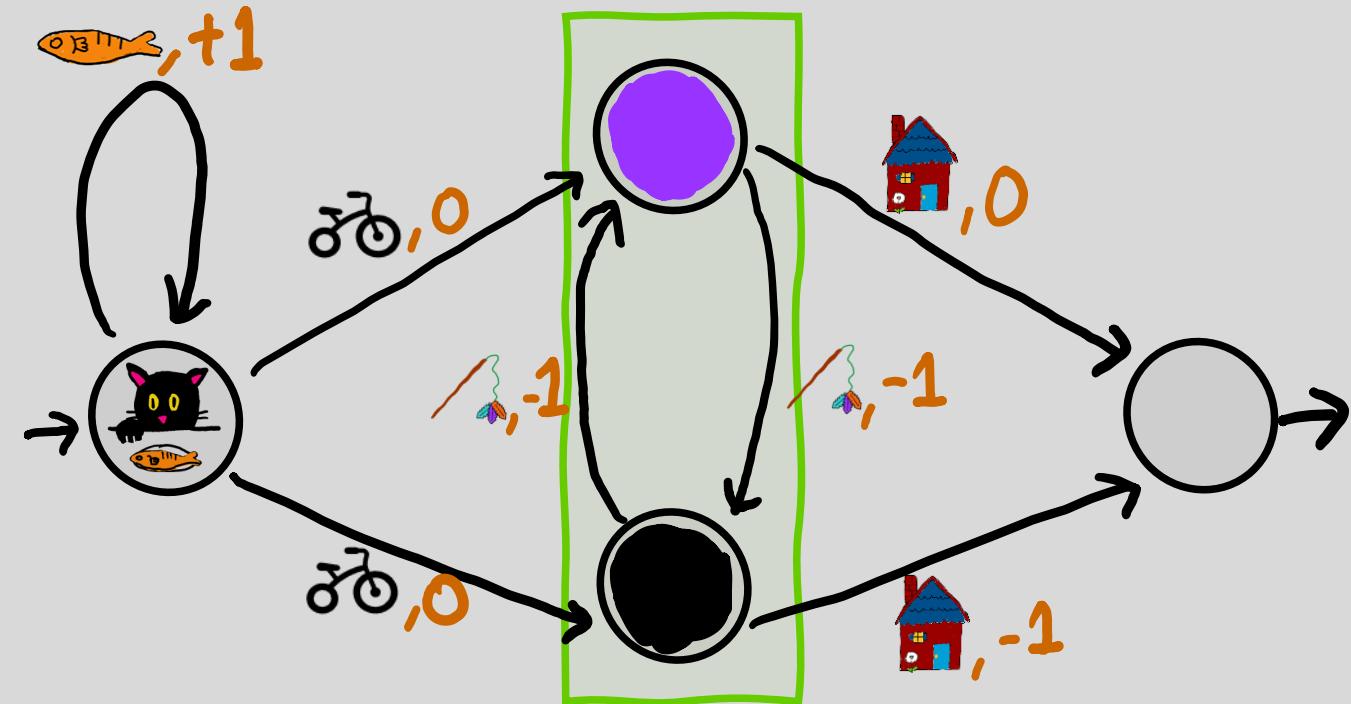


wins HD-game.

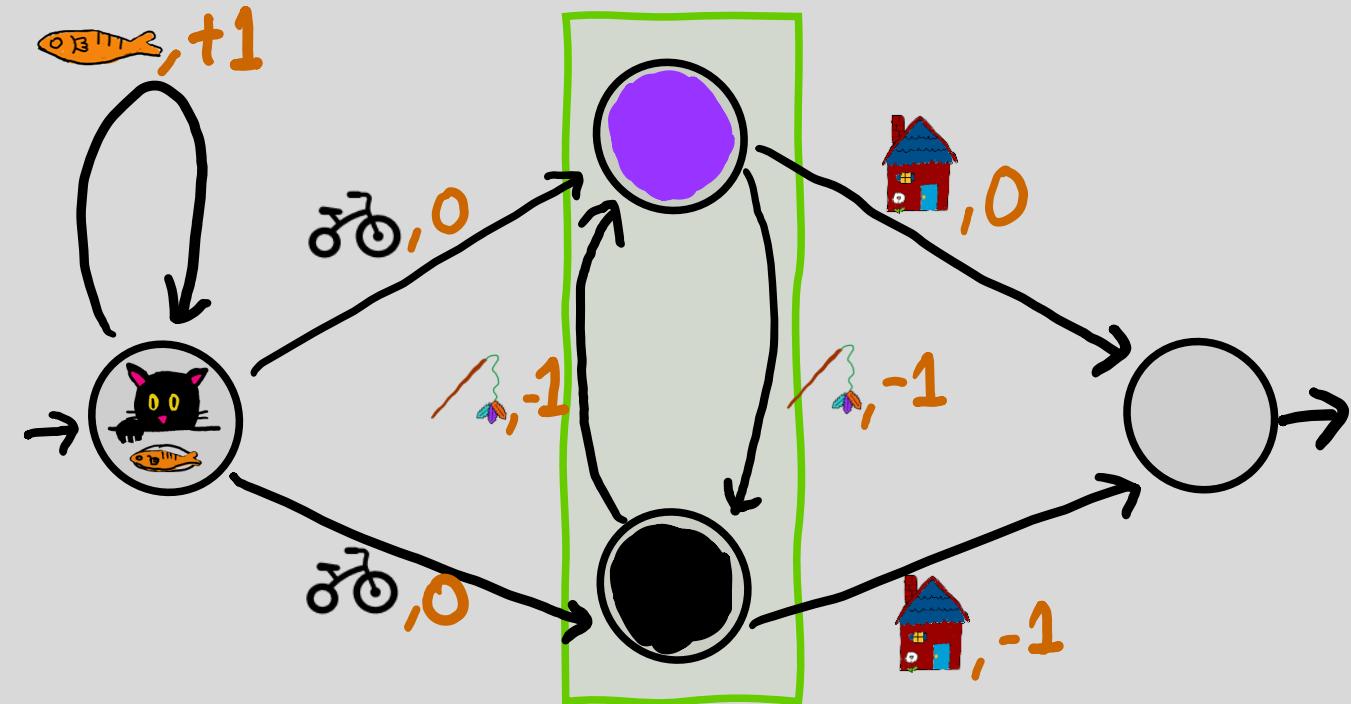
Resolver: A winning strategy of .

2. Resolvers



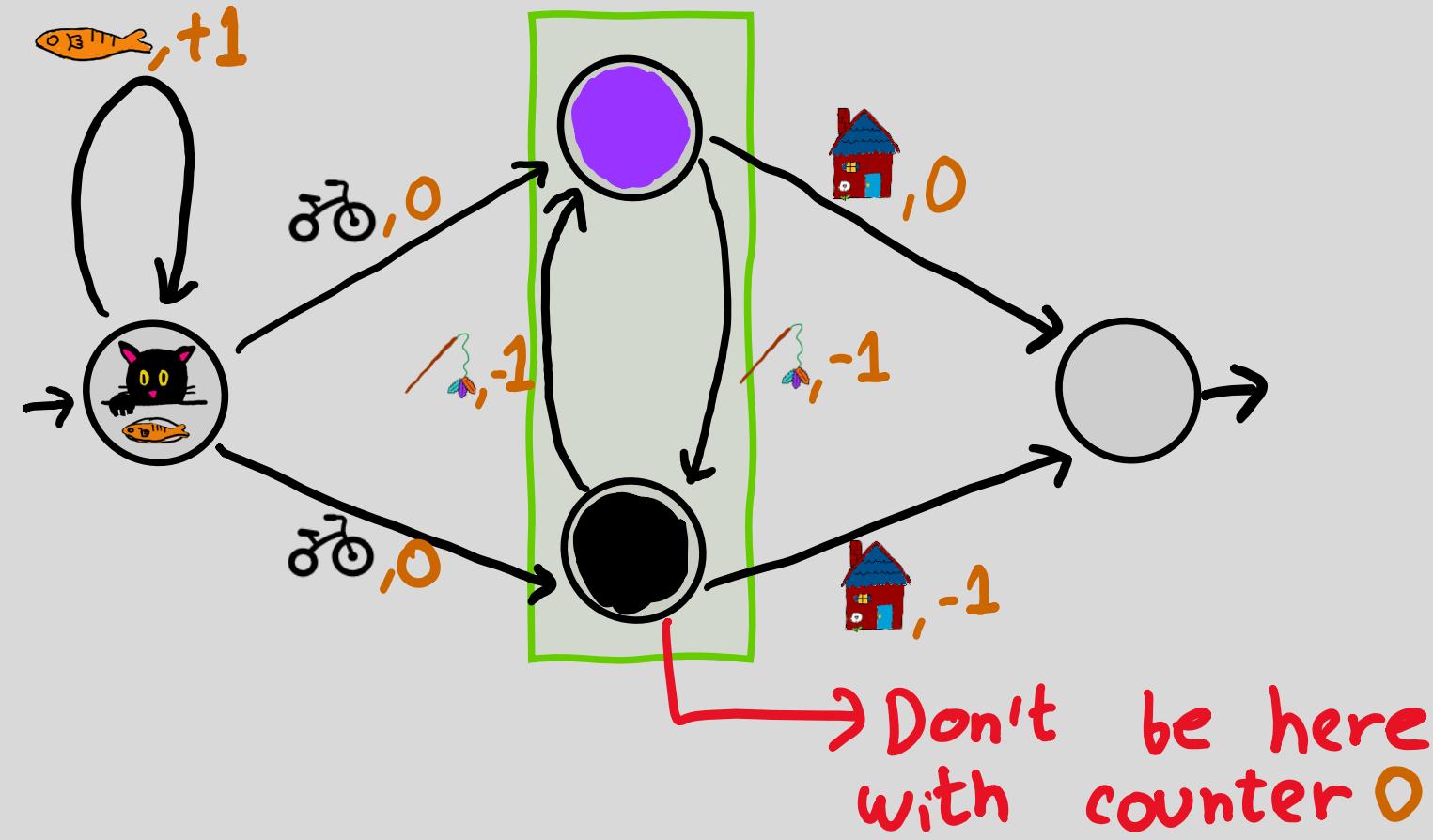


$$L = \{ \text{fish}^n \text{ } \text{bicycle}^m \text{ } \text{house}^m \mid n \geq m \}$$



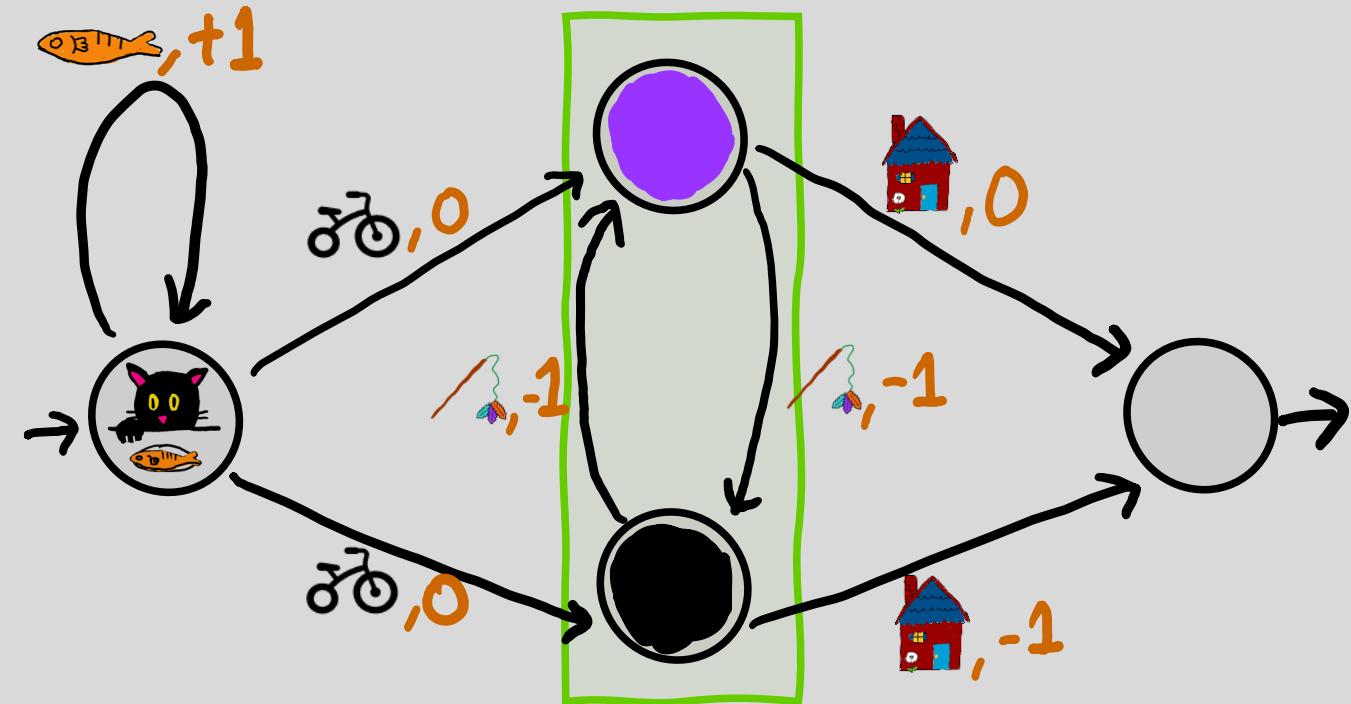
At , on :

$$L = \{ \text{fish}^n \text{ } \text{bicycle}^m \text{ } \text{house}^l \mid n \geq m \}$$



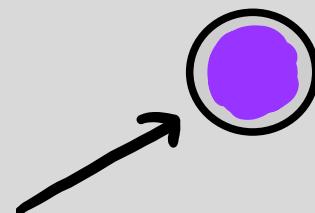
At , on :

$$L = \{ \text{Fish}^n \text{ Bike}^m \text{ House}^k \mid n \geq m \}$$



At , on :

Counter = $2n$:



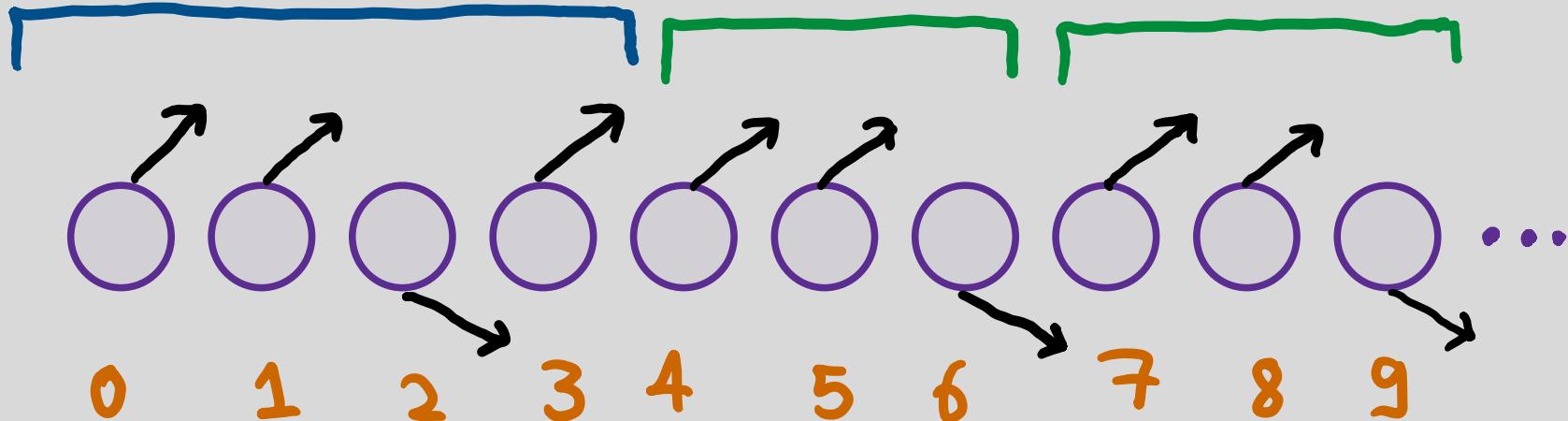
Counter = $2n + 1$:



$$L = \{ \text{fish}^n \text{ } \text{bicycle } \text{feather}^m \text{ } \text{house} \mid n \geq m \}$$

Theorem: Given an history-deterministic OCN,
there is a semilinear strategy for  that
is effectively computable.

Theorem: Given an history-deterministic OCN,
there is a semilinear strategy for  that
is effectively computable.



Theorem: Given an history-deterministic OCN,
there is a semilinear strategy for  that
is effectively computable.

Proof:



Token Game:



Token Game:



1. selects letter
2. selects transition on
3. selects transition on



Token Game:



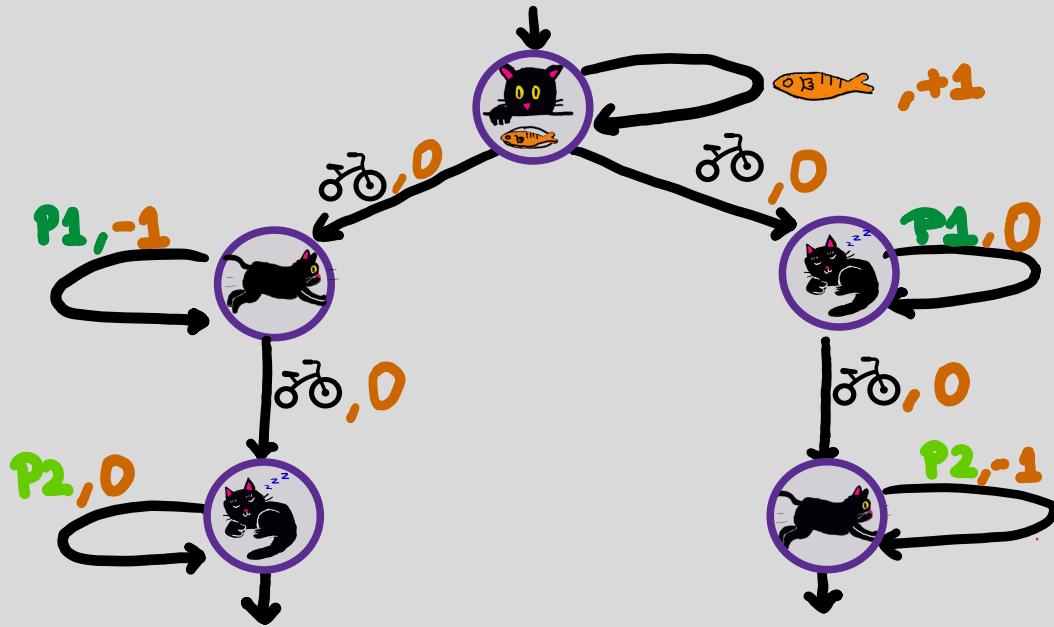
1. Devil selects letter

2. Cat selects transition on ■

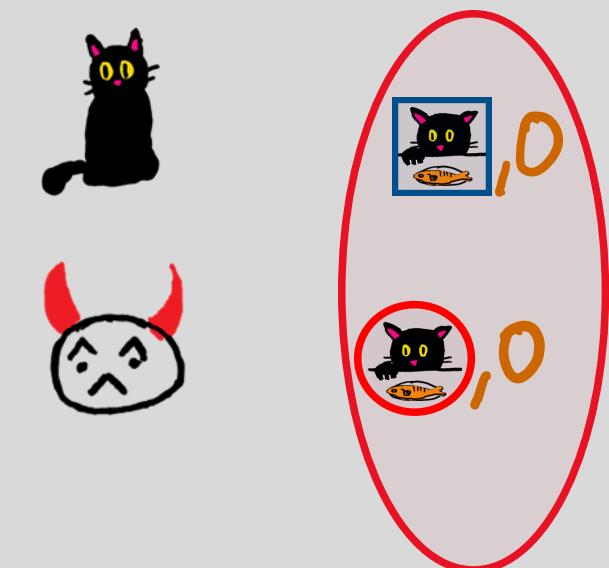
3. Devil selects transition on ●

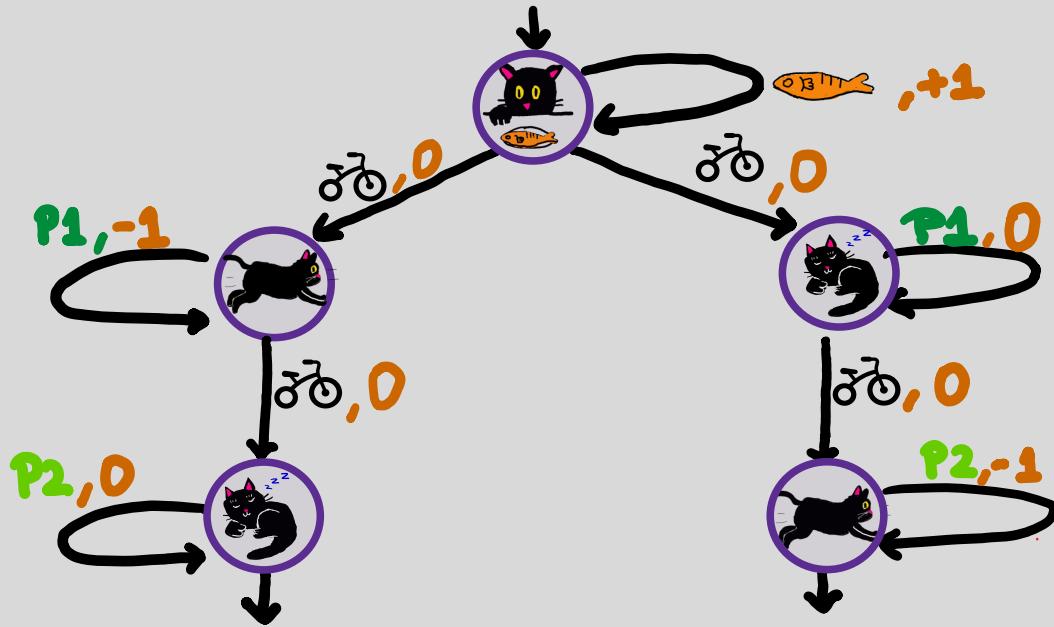


Winning condition of Devil: If Devil's run on ● is accepting and Cat's run on ■ is rejecting.

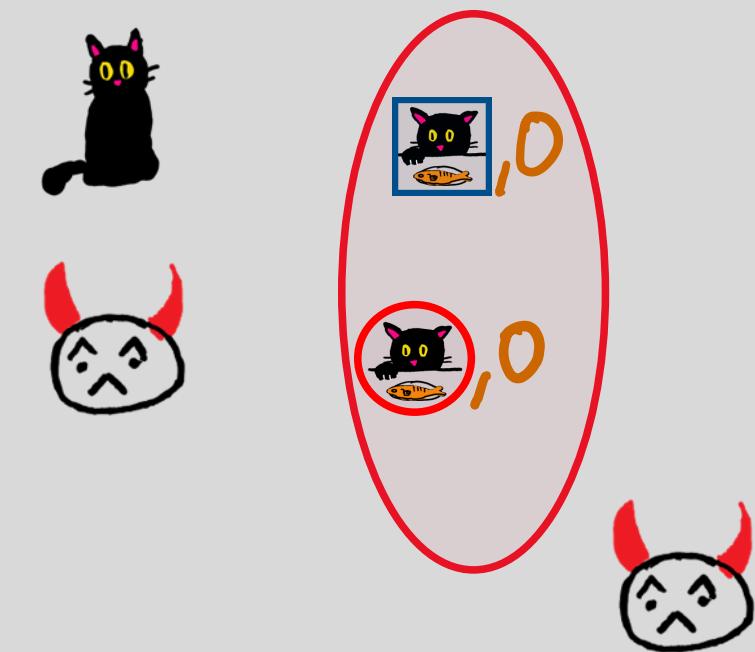


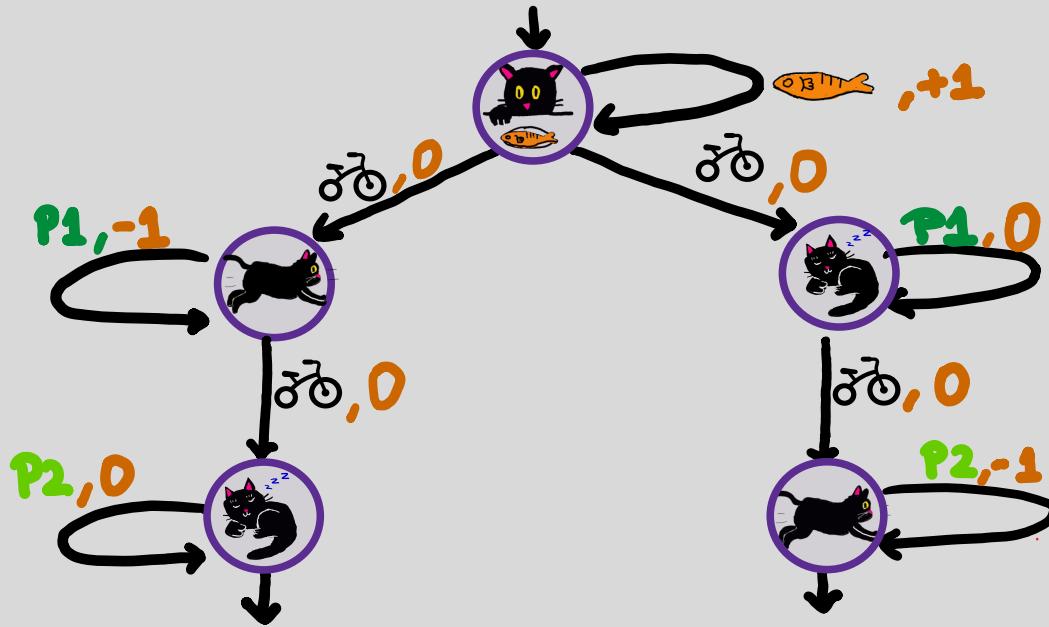
$$L = \{ \text{fish}^i \text{ } \text{bicycle } P_1^j \text{ } \text{bicycle } P_2^k \mid i \geq j \text{ or } i \geq k \}$$



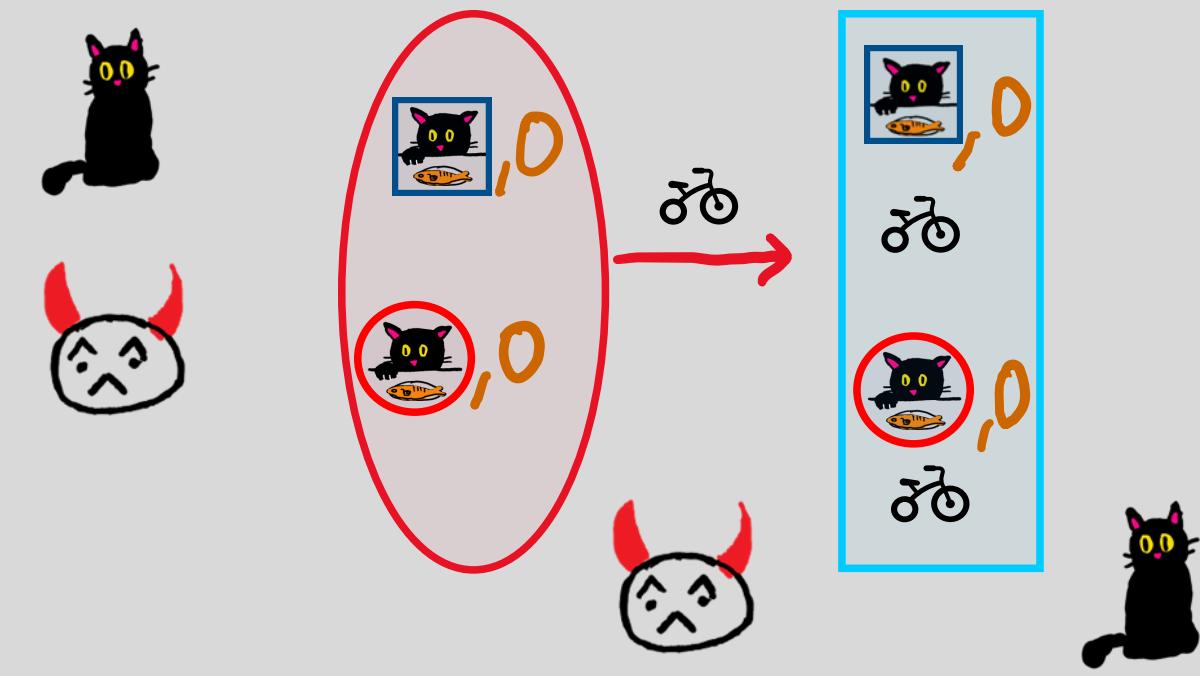


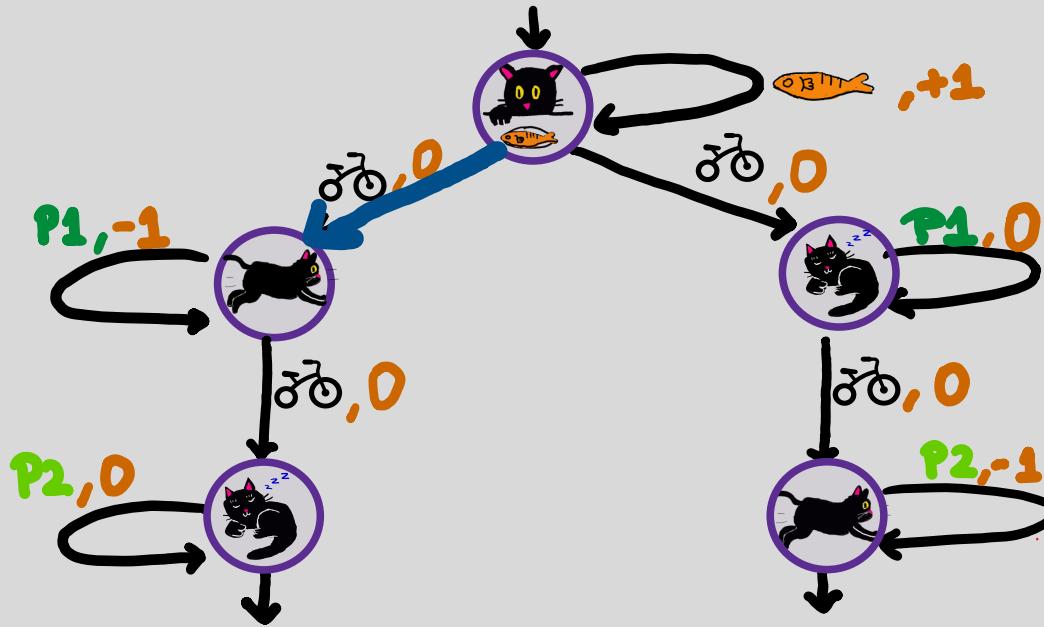
$$L = \{ \text{fish}^i \text{ } \text{bike}^{P1^j} \text{ } \text{bike}^{P2^k} \mid i \geq j \text{ or } i \geq k \}$$



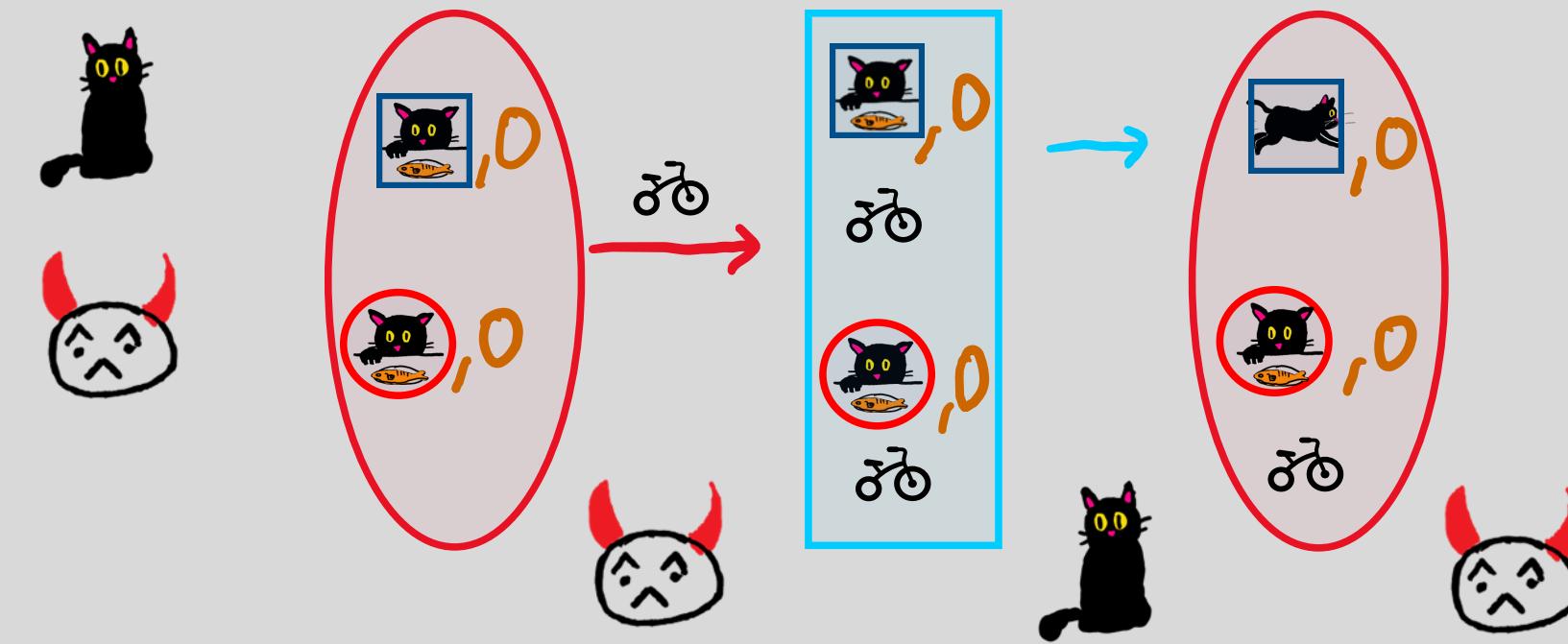


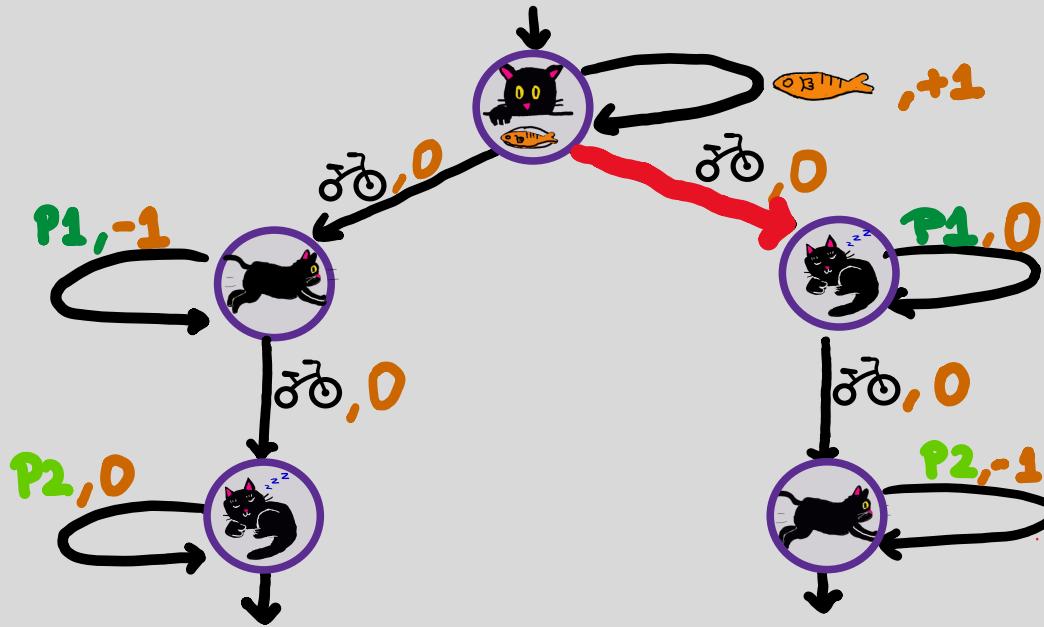
$$L = \left\{ \begin{array}{c} \text{fish}^i \\ \text{bicycle}^{P_1^j} \\ \text{bicycle}^{P_2^k} \end{array} \mid i \geq j \text{ or } i \geq k \right\}$$



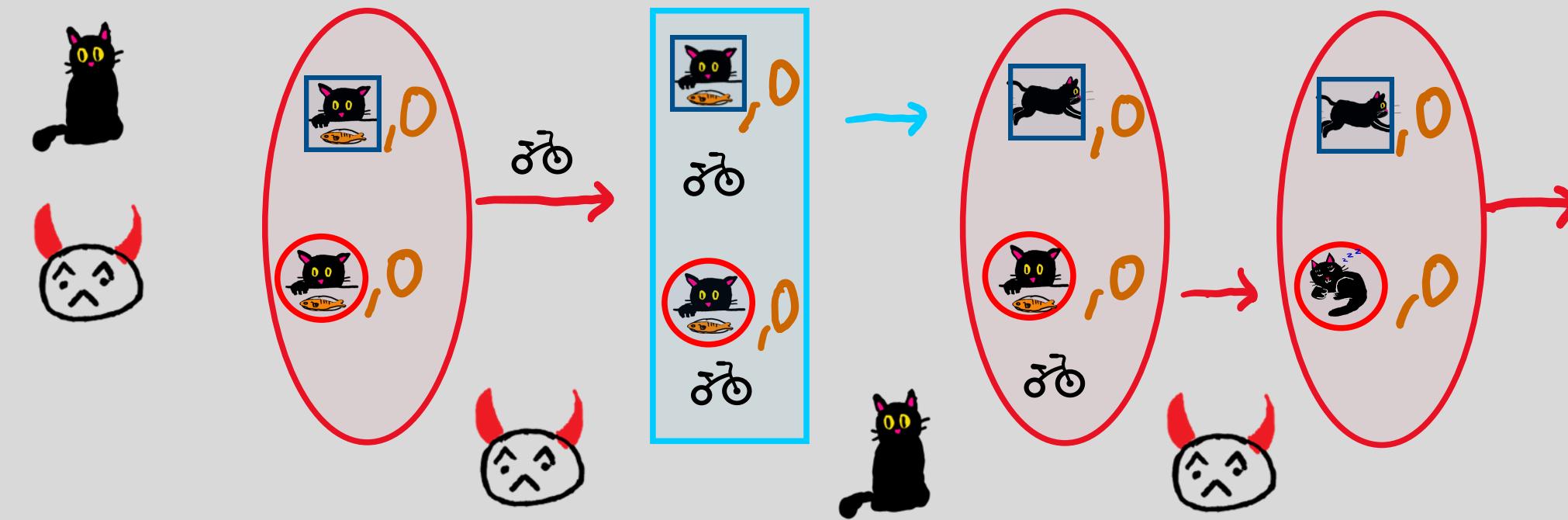


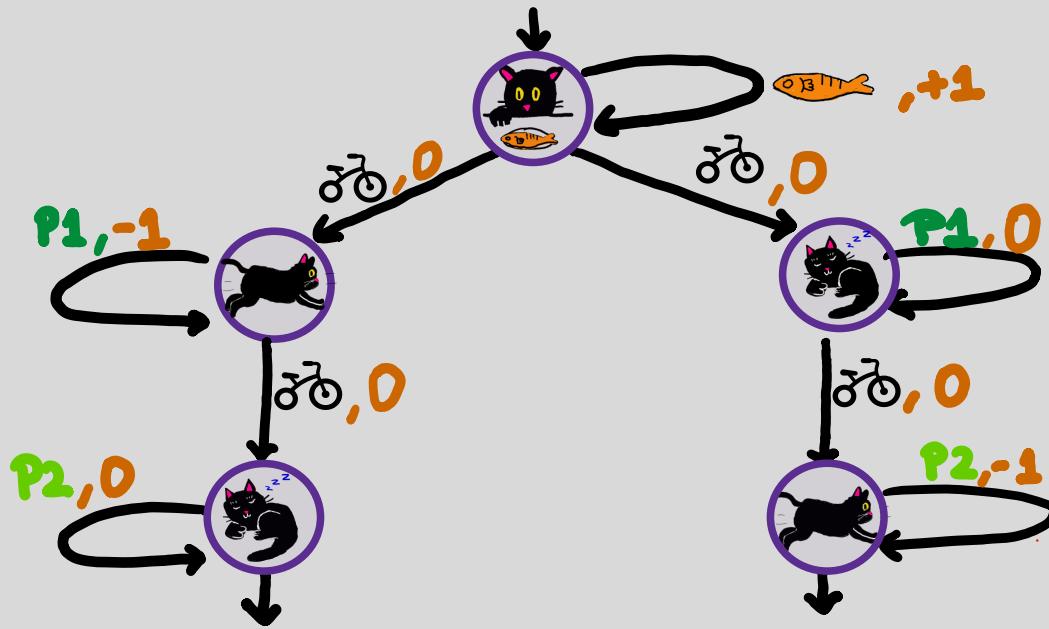
$$L = \{ \text{fish}^i \text{ } \text{bicycle}^j \text{ } \text{bicycle}^k \mid i \geq j \text{ or } i \geq k \}$$





$$L = \{ \text{fish}^i \text{ bike}^j \text{ P1}^k \text{ bike}^l \text{ P2}^m \mid i \geq j \text{ or } i \geq k \}$$





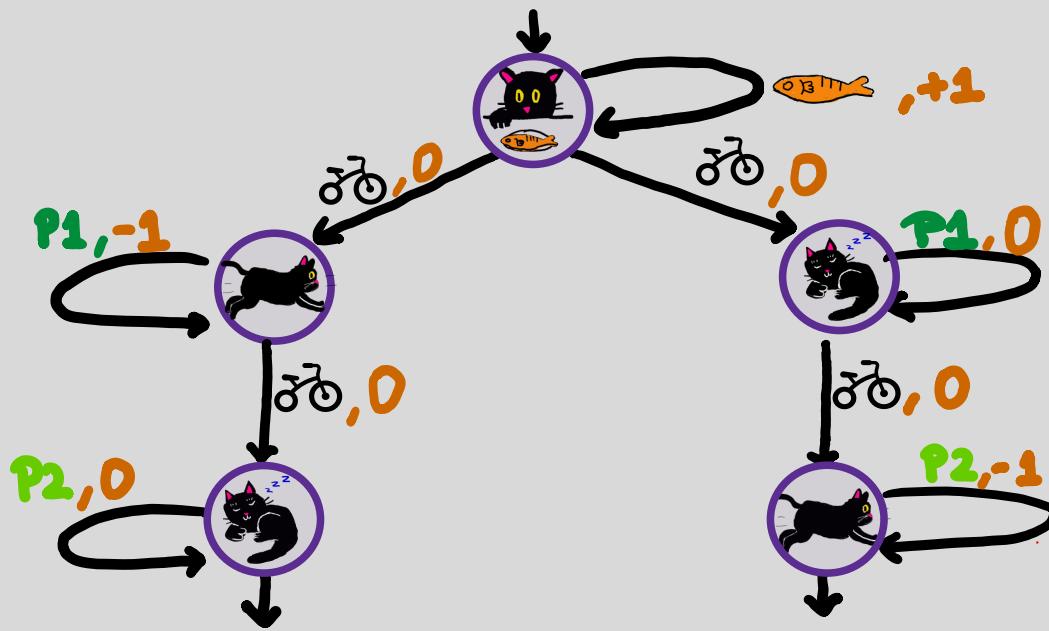
$$L = \{ \text{fish}^i \circledast P_1^j \circledast P_2^k \mid i \geq j \text{ or } i \geq k \}$$



0



0



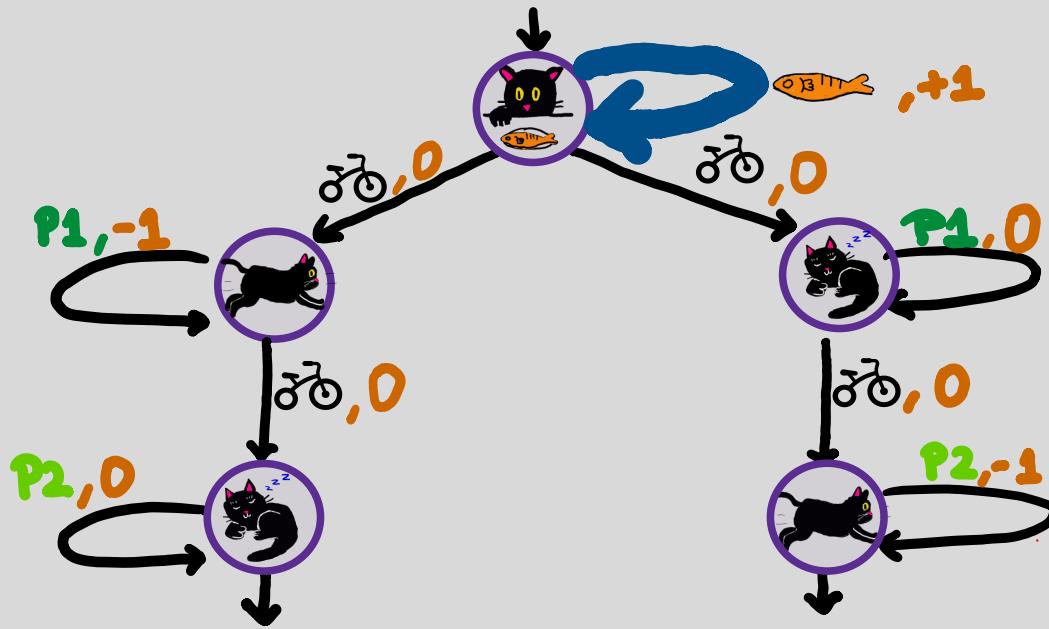
$$L = \{ \text{fish}^i \circledast P_1^j \circledast P_2^k \mid i \geq j \text{ or } i \geq k \}$$



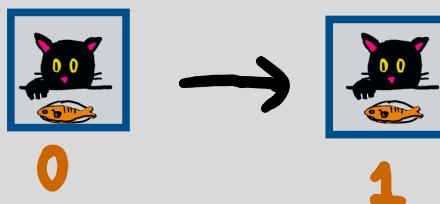
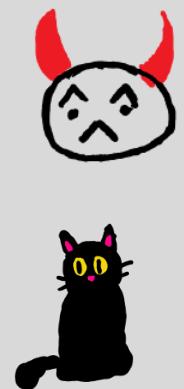
0



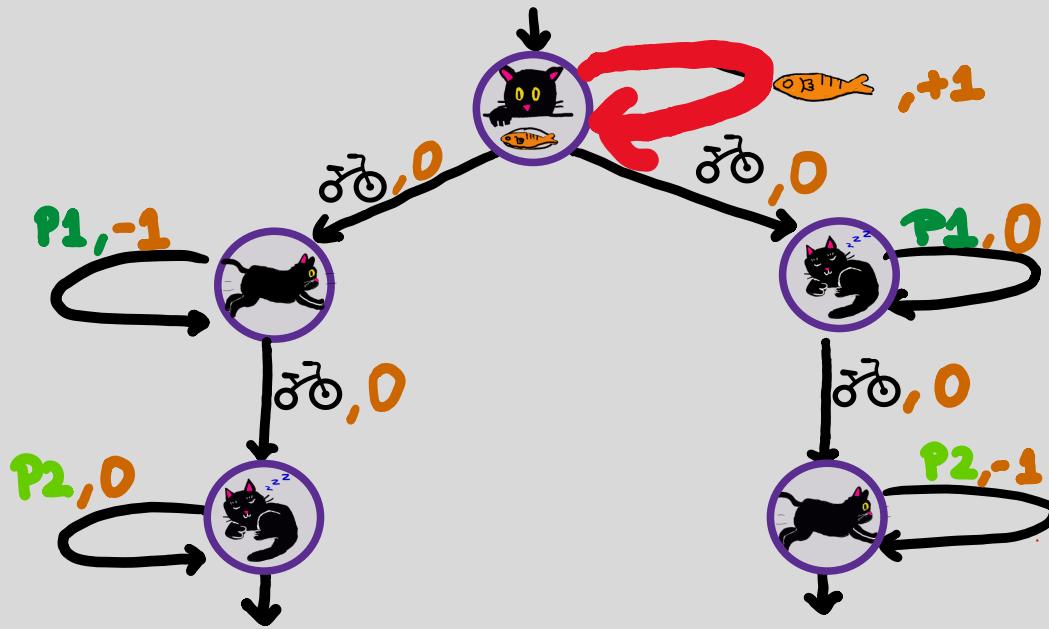
0



$$L = \{ \text{fish}^i \text{ } \text{Bicycle}^j \text{ } \text{Bicycle}^k \mid i \geq j \text{ or } i \geq k \}$$



0



$$L = \left\{ \text{fish}^i \text{ bike } P_1^j \text{ bike } P_2^k \mid i \geq j \text{ or } i \geq k \right\}$$



0



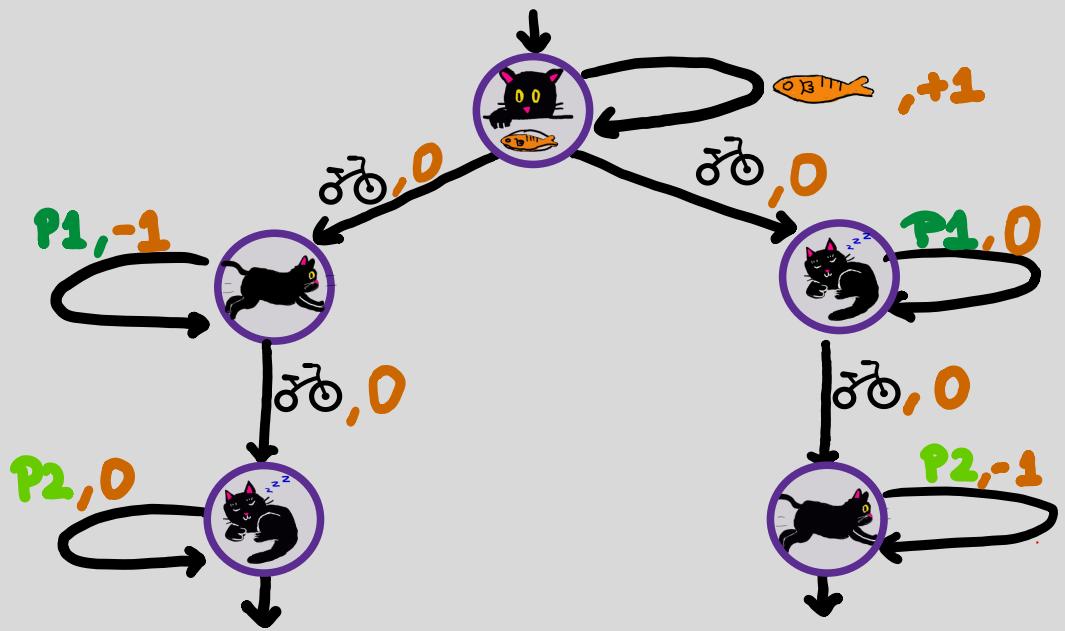
1



0

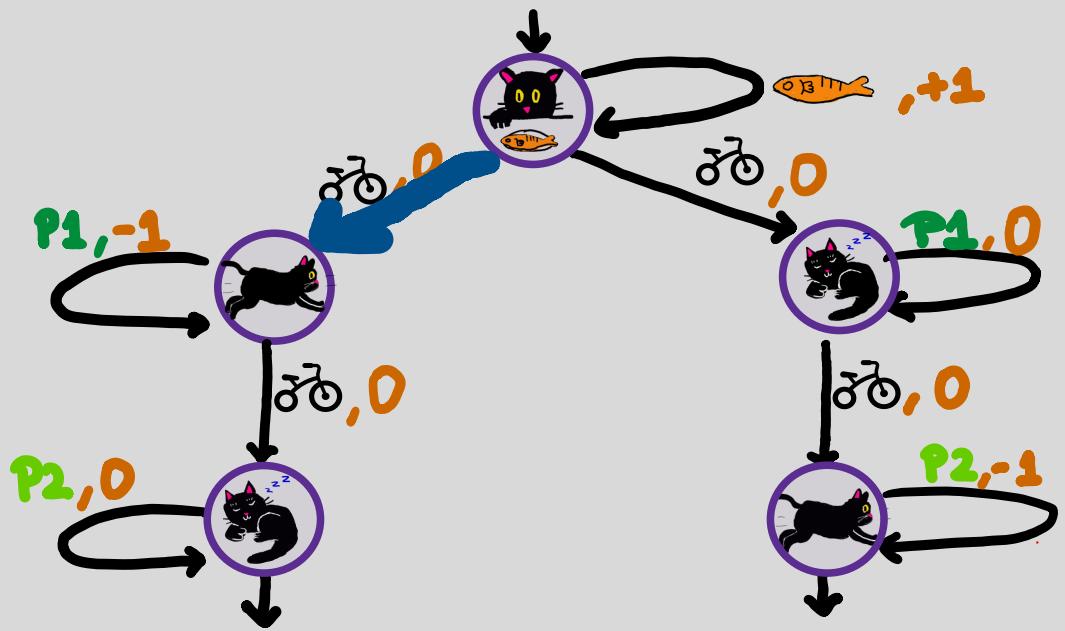


1



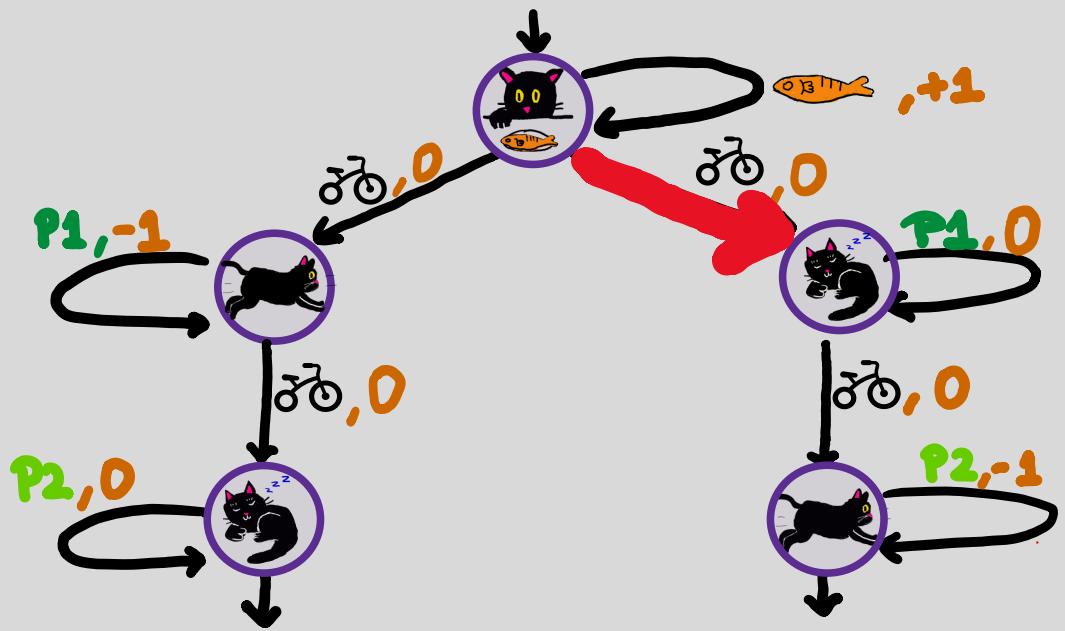
$$L = \{ \text{fish}^i \text{ } \text{bicycle}^j \text{ } \text{bicycle}^k \mid i \geq j \text{ or } i \geq k \}$$





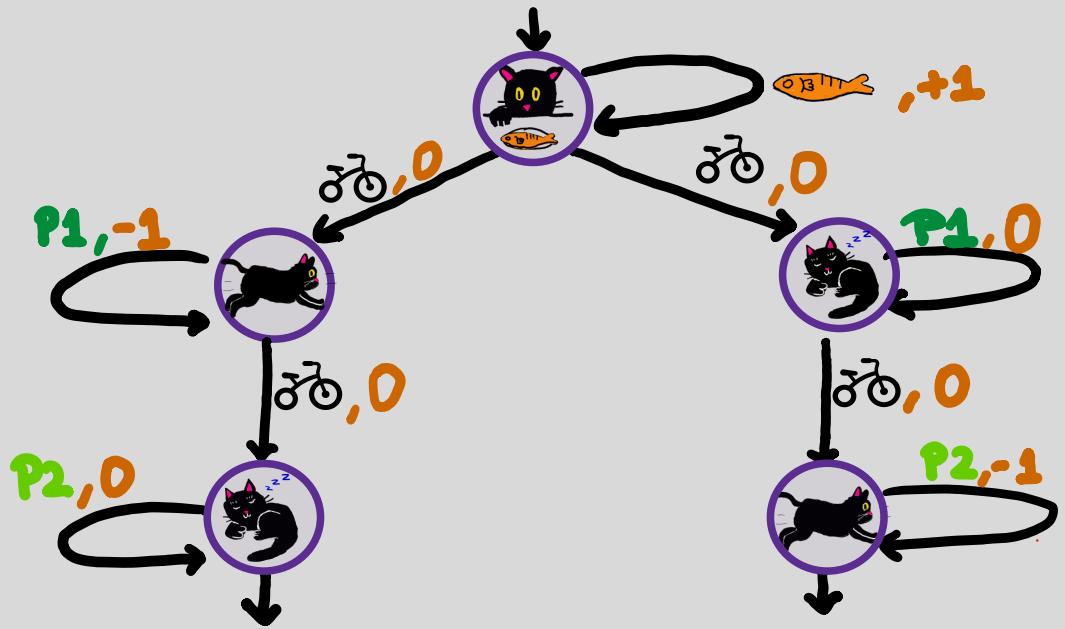
$$L = \{ \text{carrot}^i \text{ bike } P_1^j \text{ bike } P_2^k \mid i \geq j \text{ or } i \geq k \}$$



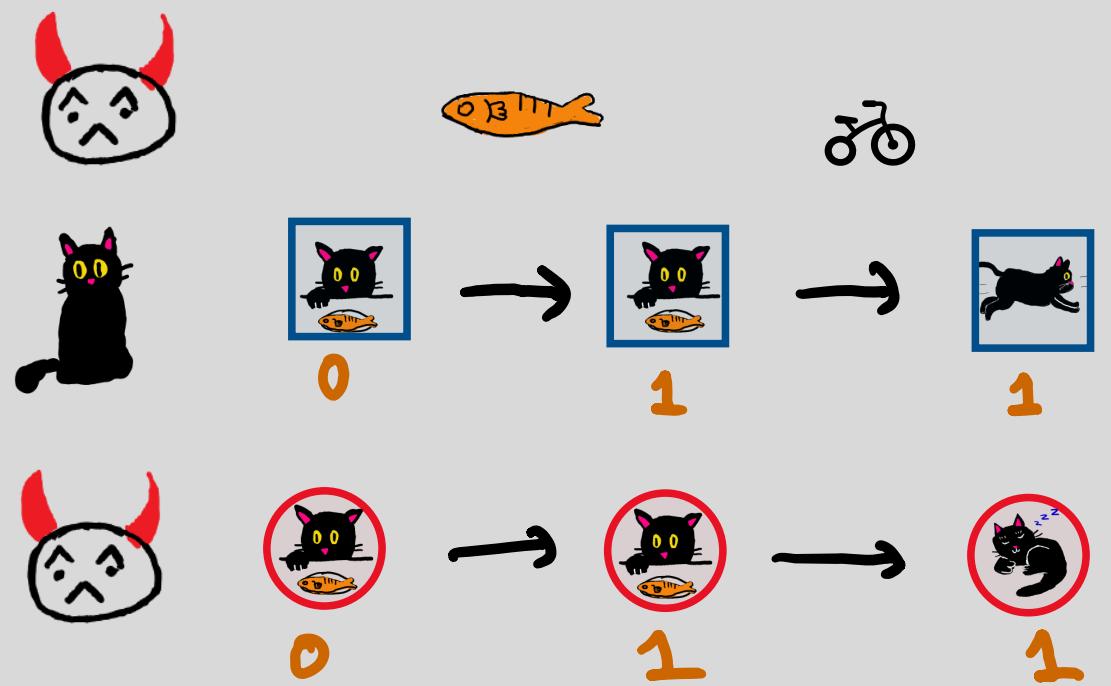


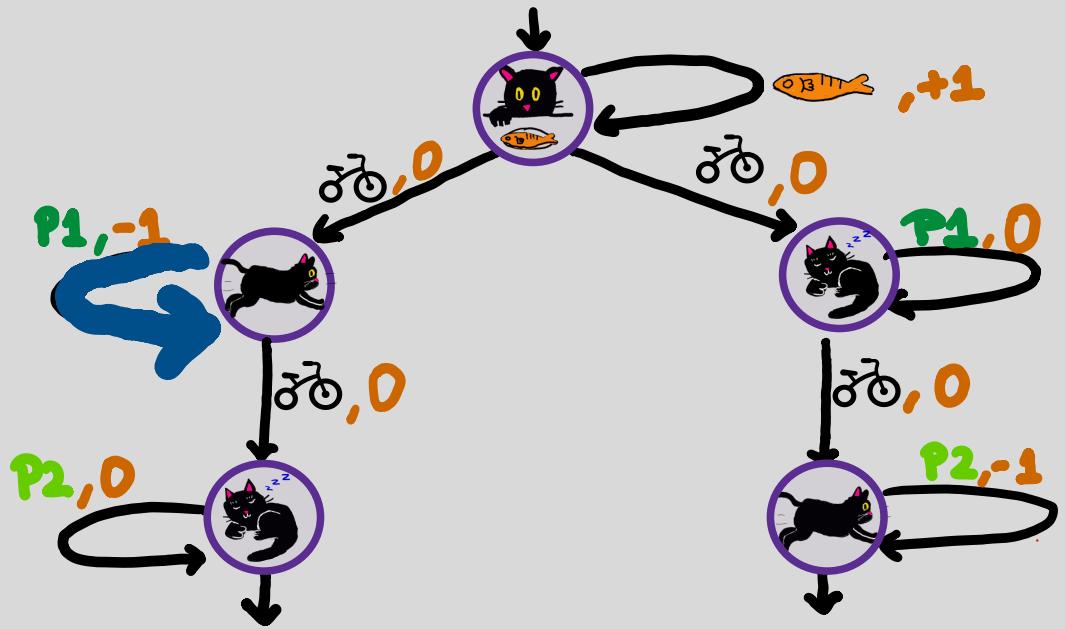
$$L = \{ \text{fish}^i \text{ } \text{bicycle}^j \text{ } \text{bicycle}^k \mid i \geq j \text{ or } i \geq k \}$$





$$L = \left\{ \text{fish}^i \text{ } \text{bicycle}^j \text{ } \text{fish}^k \mid i \geq j \text{ or } i \geq k \right\}$$



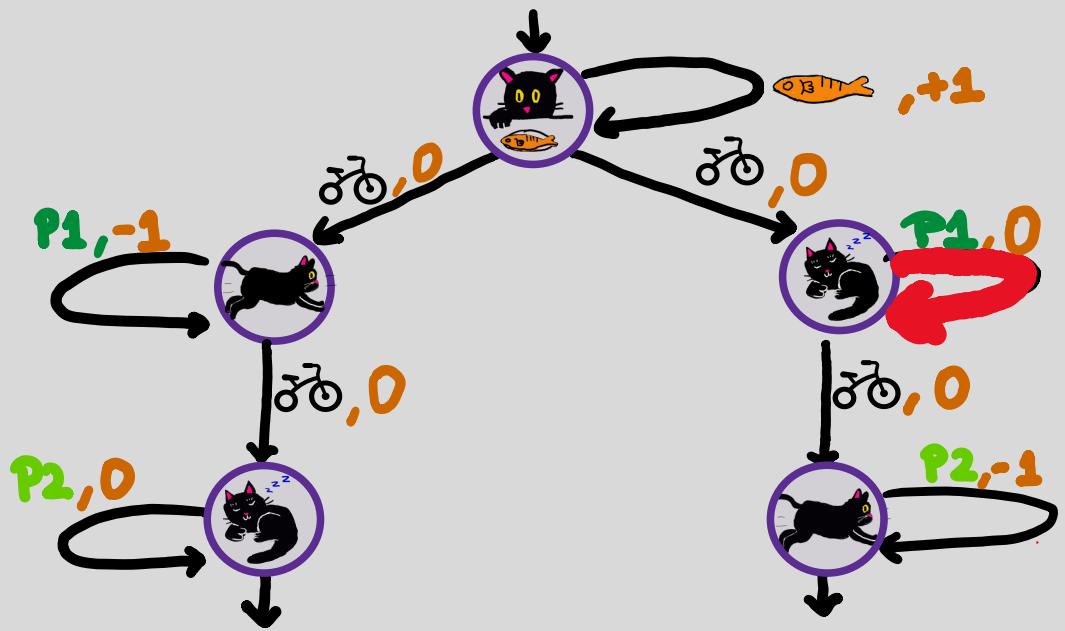


$$L = \{ \underbrace{\text{fish}}_i \xrightarrow{\text{bicycle}} \underbrace{P1^j}_i \xrightarrow{\text{bicycle}} \underbrace{P2^k}_i \mid i \geq j \text{ or } i \geq k \}$$

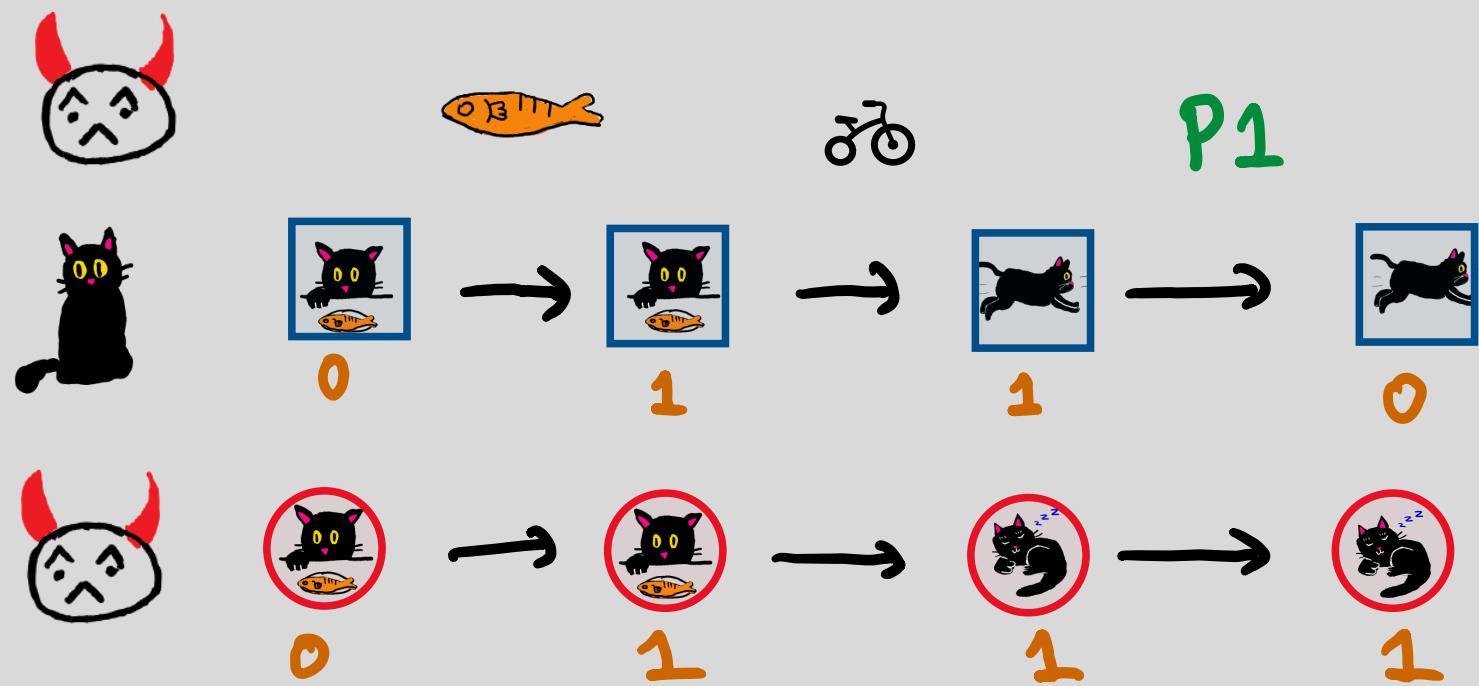


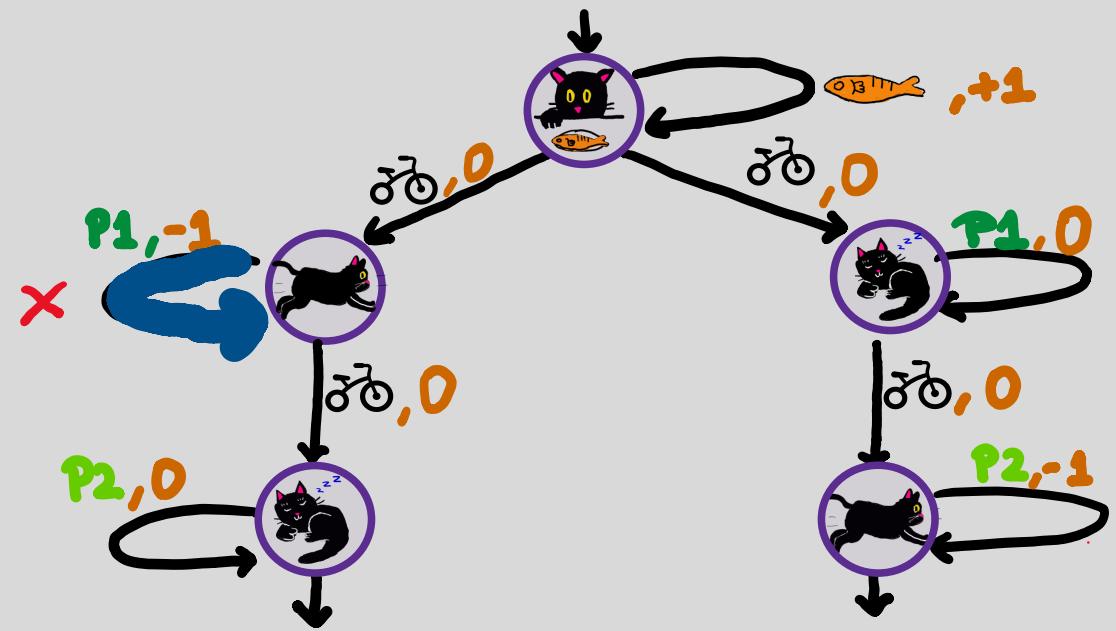
$P1$





$$L = \left\{ \text{fish}^i \text{ bike}^j \text{ P1}^k \text{ P2}^l \mid i \geq j \text{ or } i \geq k \right\}$$





$$L = \{ \text{fish}^i \text{ } \text{bicycle}^j \text{ } \text{bicycle}^k \mid i \geq j \text{ or } i \geq k \}$$



P1

P1



0



1



1



0



0



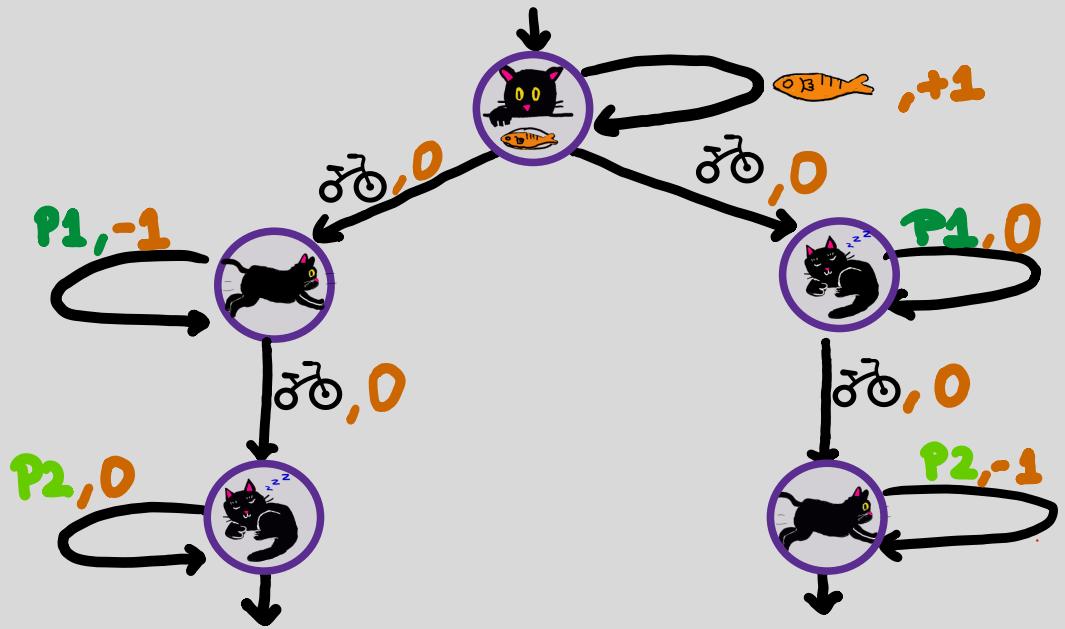
1



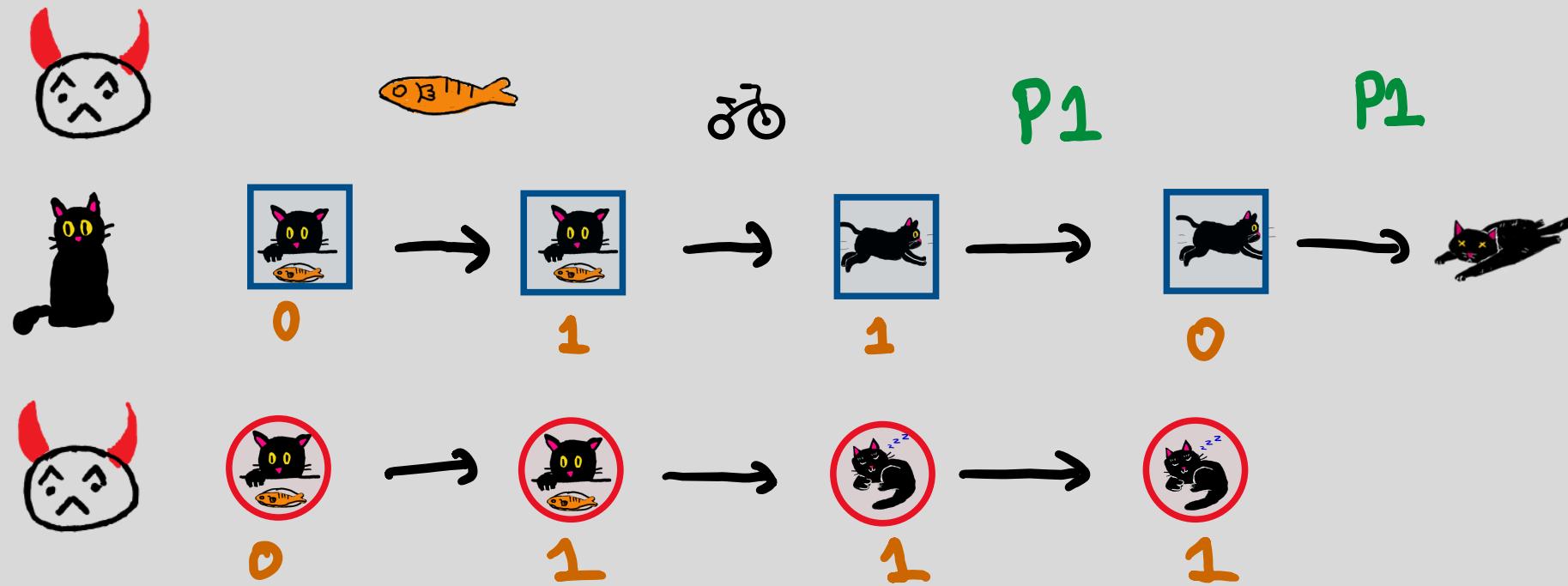
1

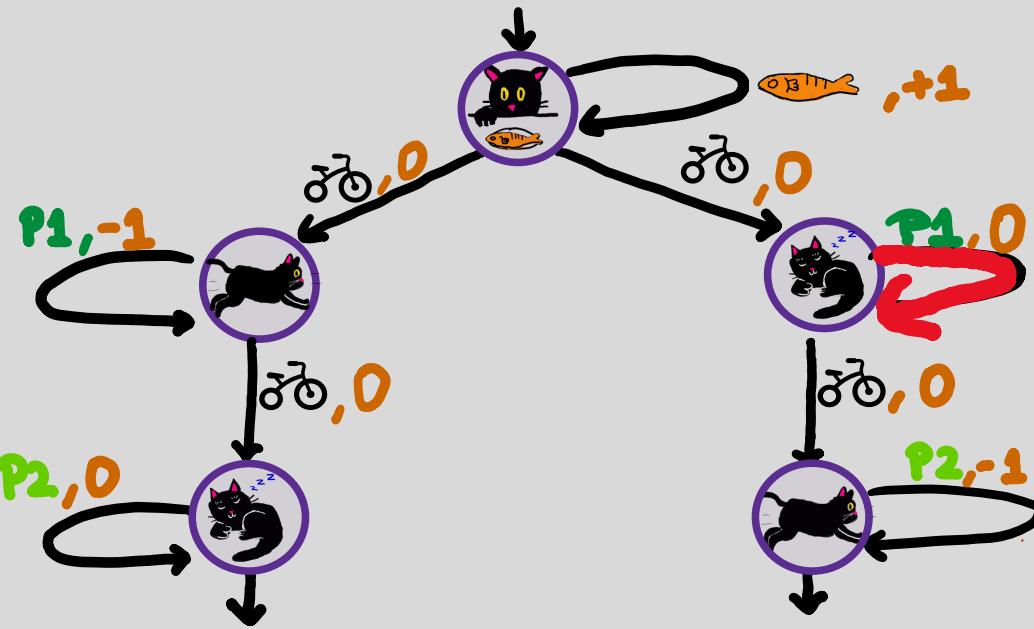


1

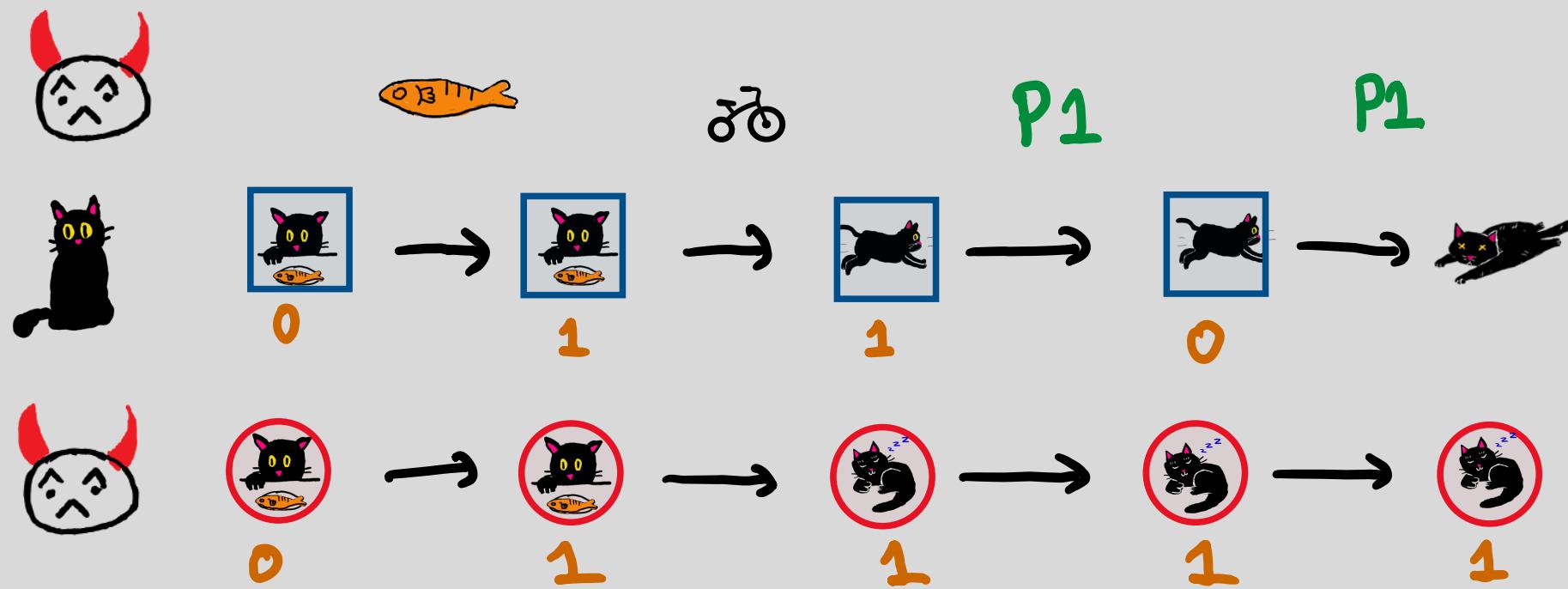


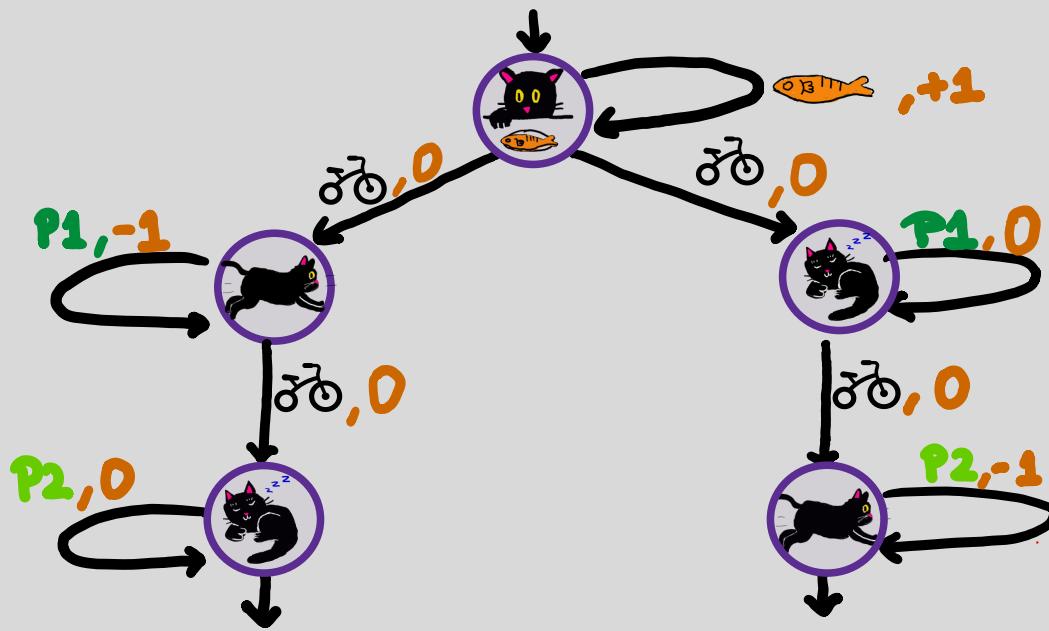
$$L = \left\{ \overbrace{\text{fish}}^i \xrightarrow{\text{bicycle}} P1^j \xrightarrow{\text{bicycle}} P2^k \mid i \geq j \text{ or } i \geq k \right\}$$



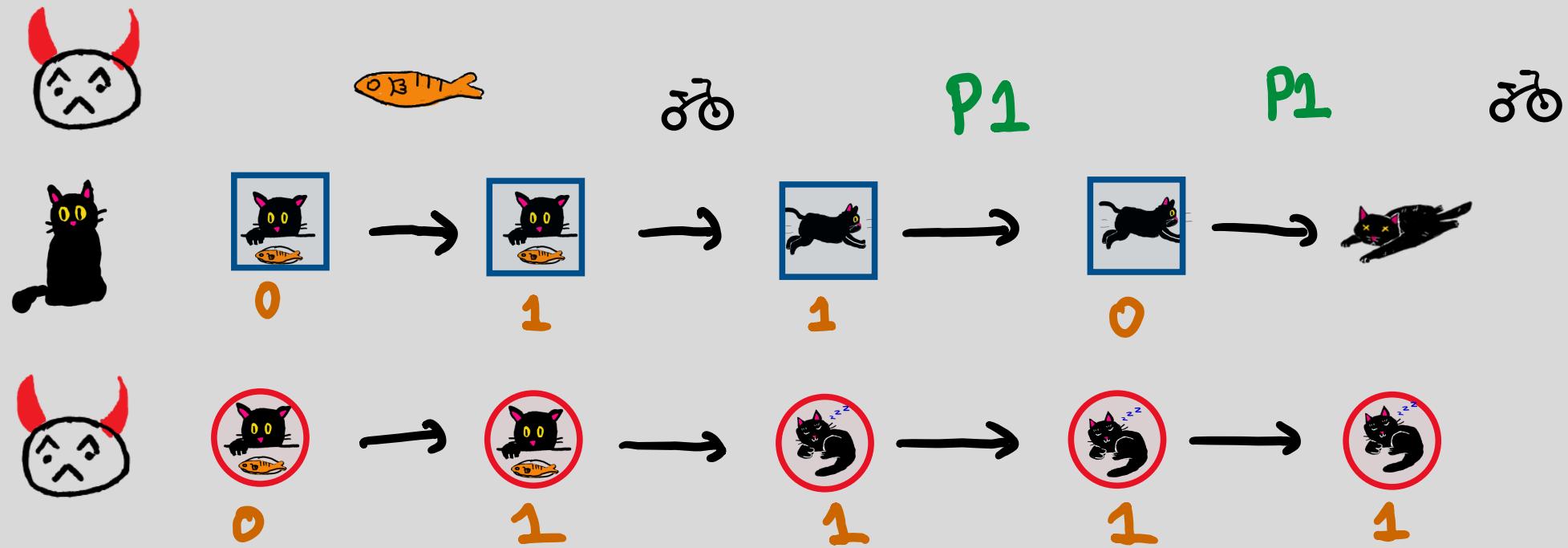


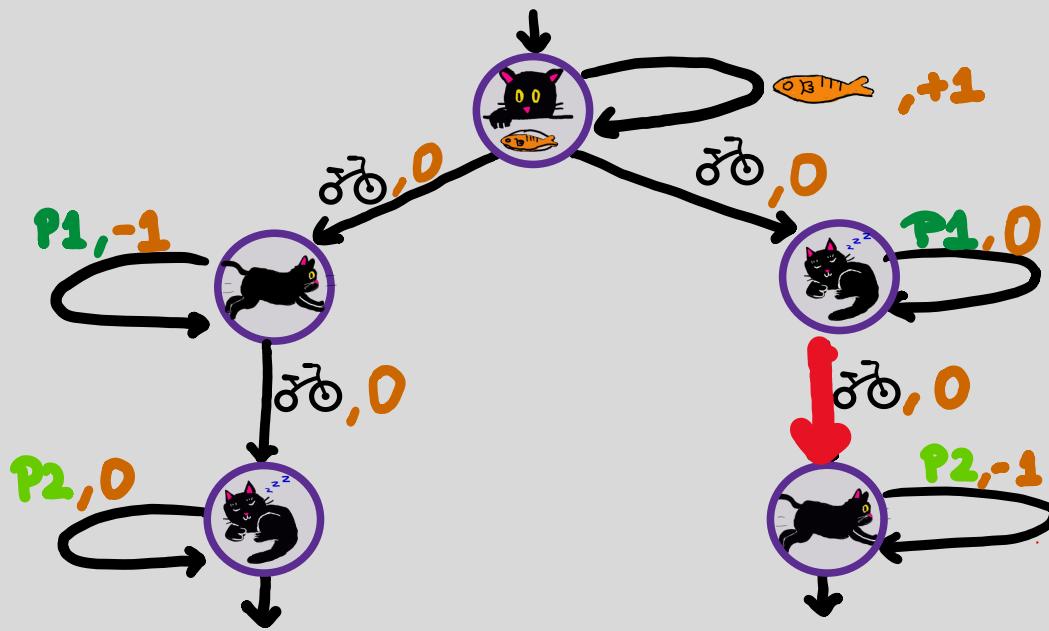
$$L = \{ \text{fish}^i \text{ } \text{bike}^j \text{ } \text{bike}^k \mid i \geq j \text{ or } i \geq k \}$$



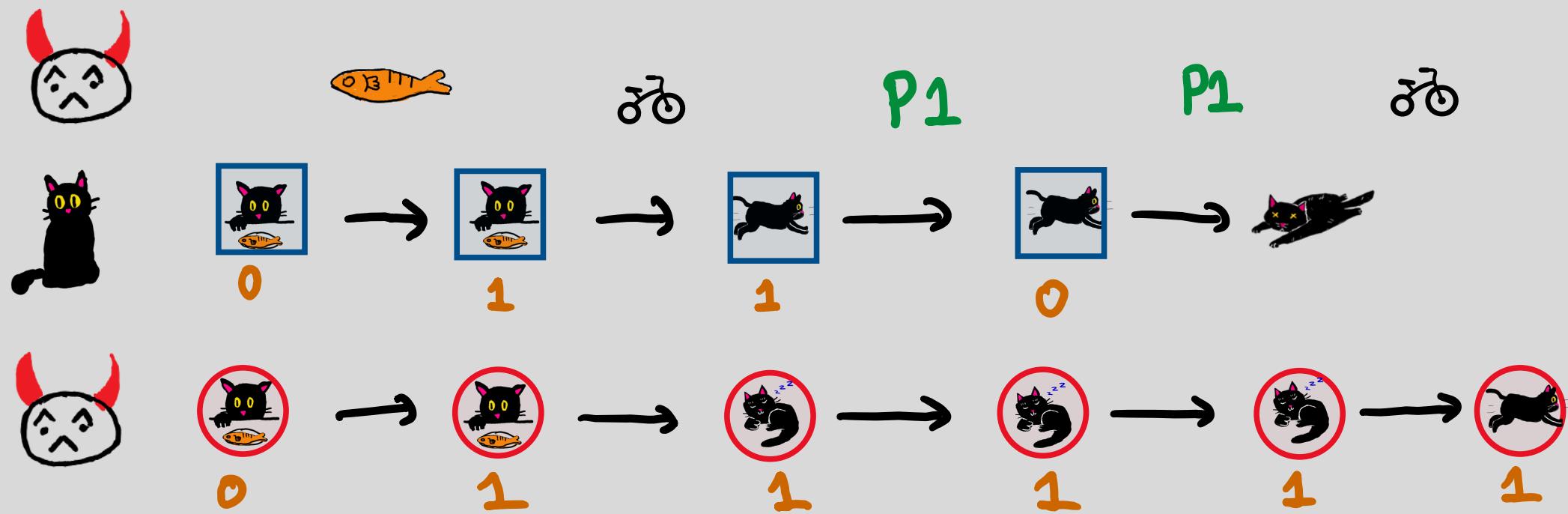


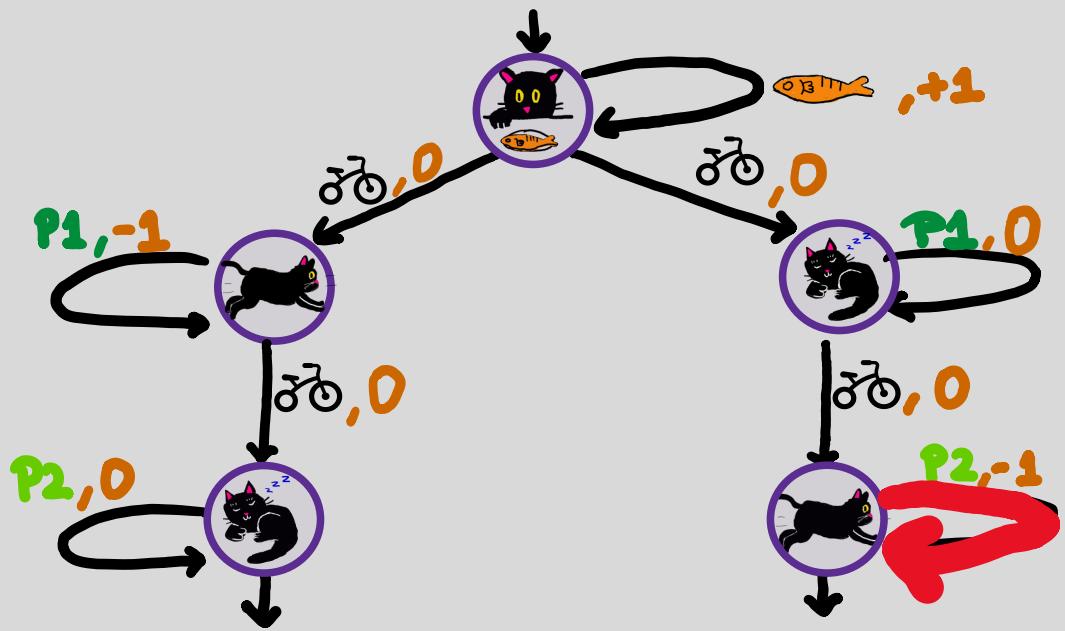
$$L = \{ \underbrace{\text{fish}}_i \xrightarrow{\text{bicycle}} \underbrace{P1}_j \xrightarrow{\text{bicycle}} \underbrace{P2}_k \mid i \geq j \text{ or } i \geq k \}$$



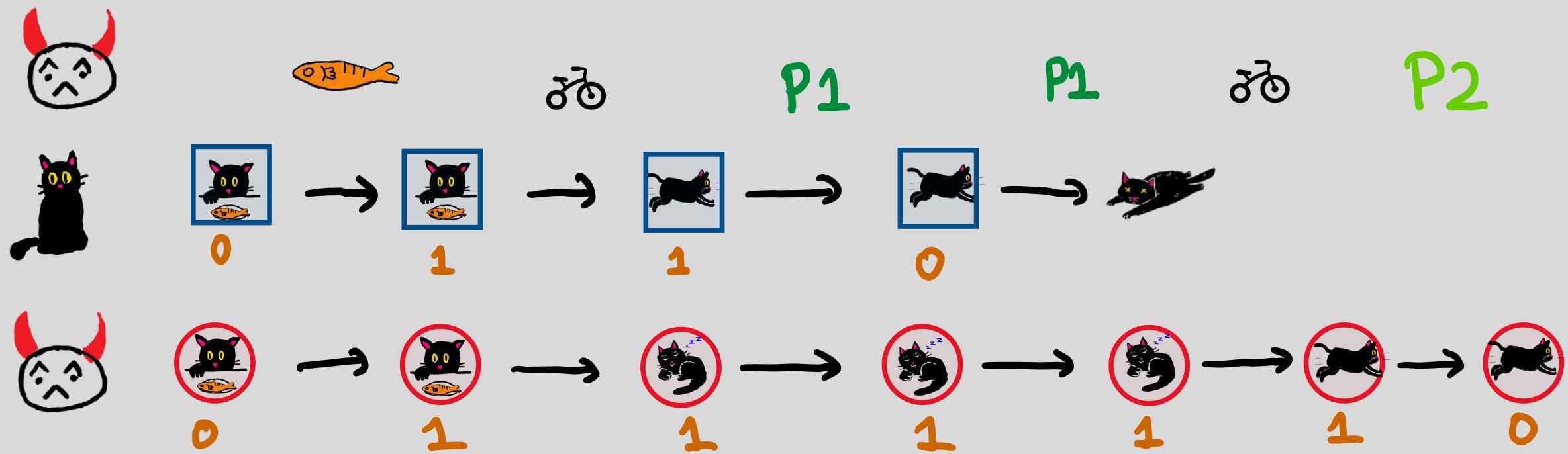


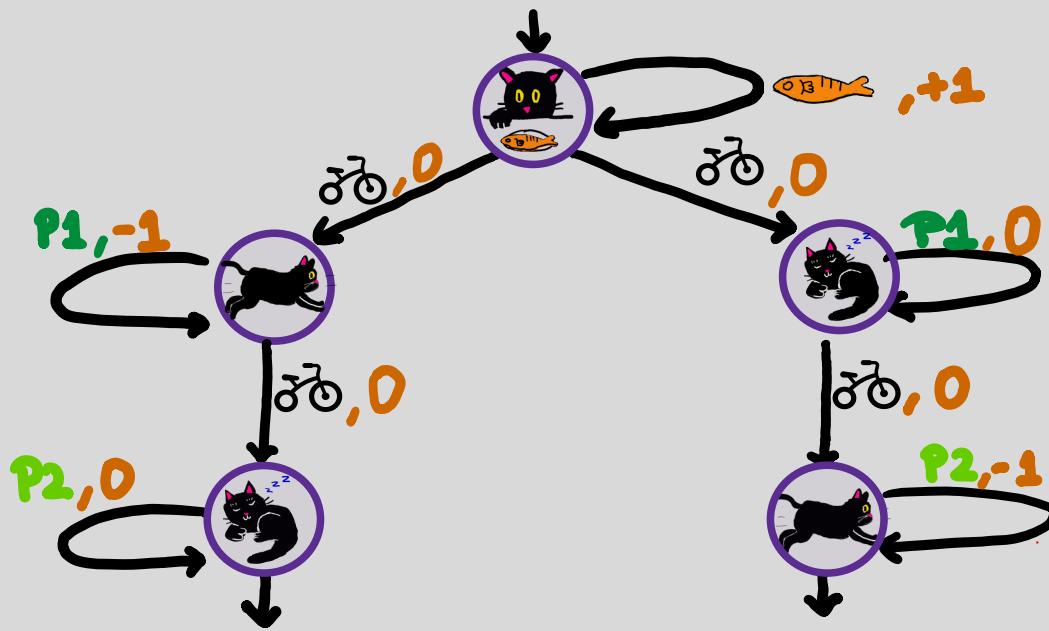
$$L = \{ \text{fish}^i \text{ } \text{bicycle}^j \text{ } \text{bicycle}^k \mid i \geq j \text{ or } i \geq k \}$$





$$L = \left\{ \text{fish}^i \text{ bike}^{P1^j} \text{ bike}^{P2^k} \mid i \geq j \text{ or } i \geq k \right\}$$





$$L = \{ \underbrace{\text{fish}}_i \xrightarrow{\text{B}} \underbrace{\text{P1}}^j \xrightarrow{\text{B}} \underbrace{\text{P2}}^k \mid i \geq j \text{ or } i \geq k \}$$



P1

P1



P2



0



1



1



0



X



0



1



1



1



1



0

✓

HD Game:

Winning Condition for : If 's word is accepting,
and 's run is rejecting.

Token Game:

Winning condition of  : If 's run on  is
accepting and 's run on  is rejecting.

Theorem [Boker, Lehtinen '22]



wins 1-token game \Leftrightarrow wins HD-game

Theorem: Given an history-deterministic OCN,
there is a semilinear strategy for  that
is effectively computable.

Proof:



Theorem [Hofman, Lasota, Mayr, Totzke' 16]

Deciding the winner in the simulation game between two OCNs can be done in PSPACE, and the winning player has semilinear strategies.

Proof:

HD
Game



Token
Game



Simulation
Game

Semilinear
Strategies

Proof:

HD
Game



Token
Game



Simulation
Game

Semilinear ←
Strategies

Semilinear
Strategies

Proof:

HD
Game



Token
Game



Simulation
Game

Semilinear
Strategies

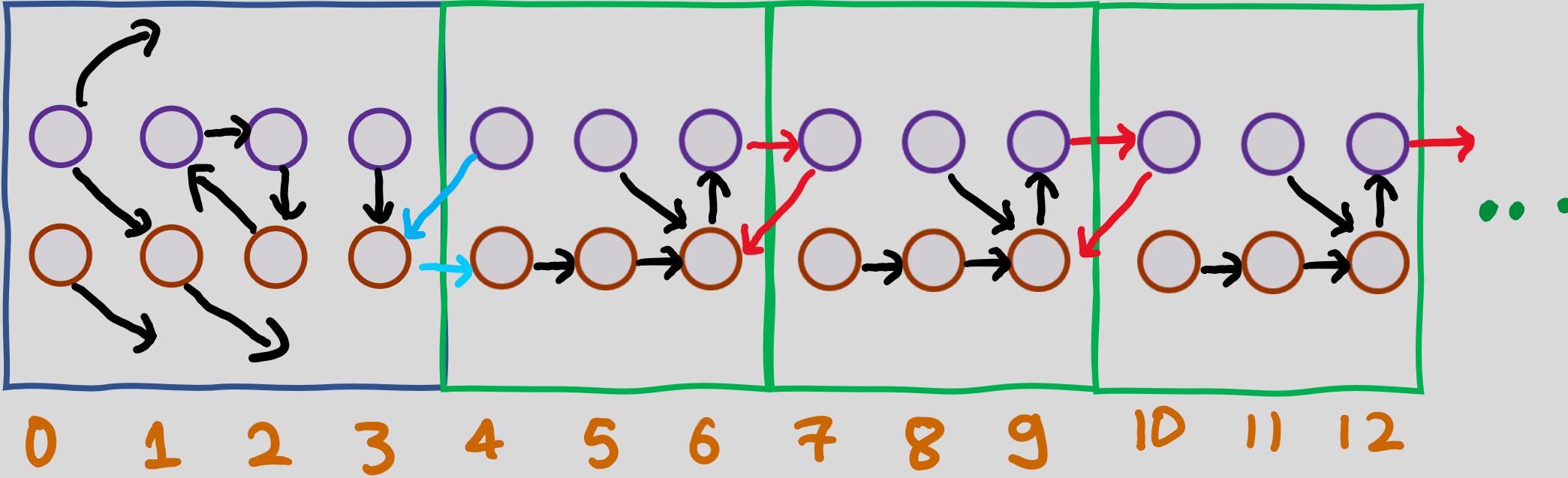


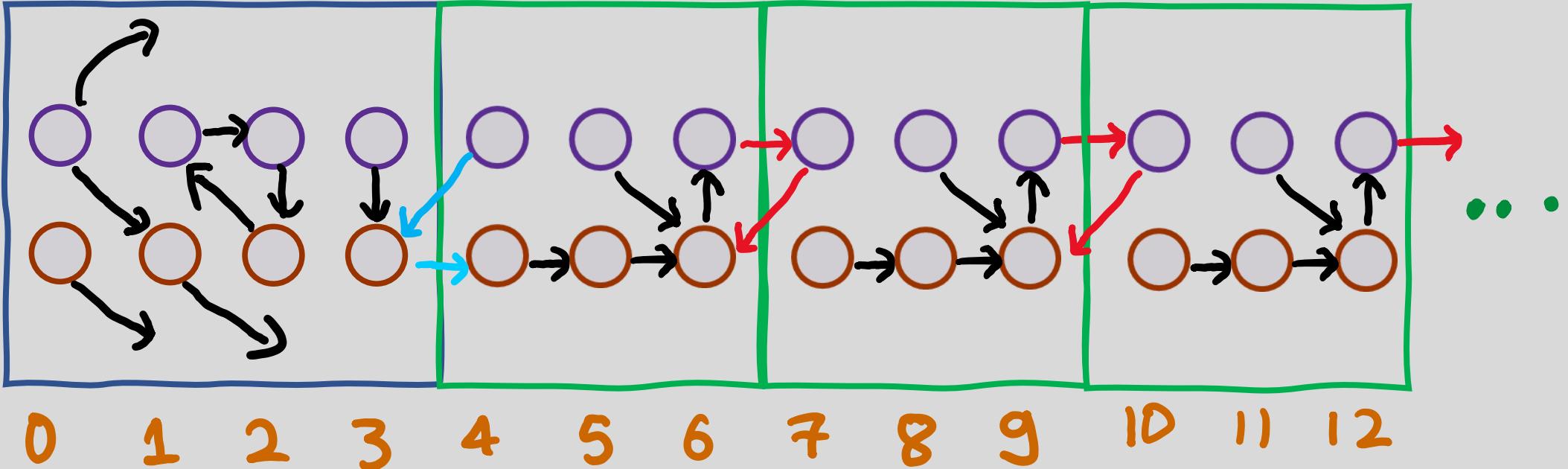
Semilinear
Strategies



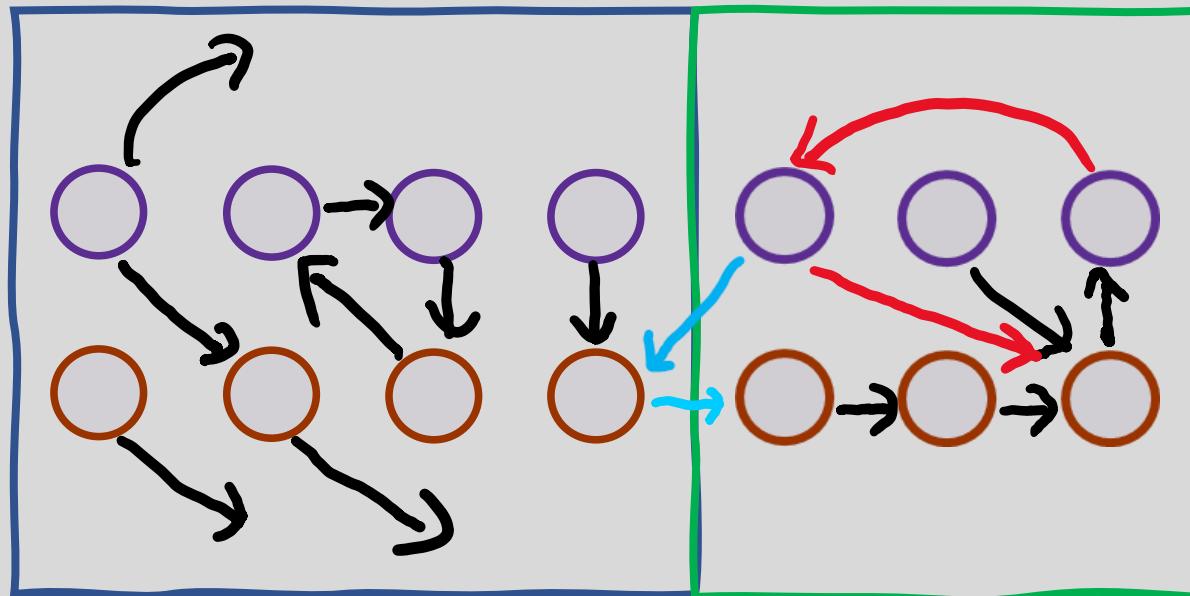
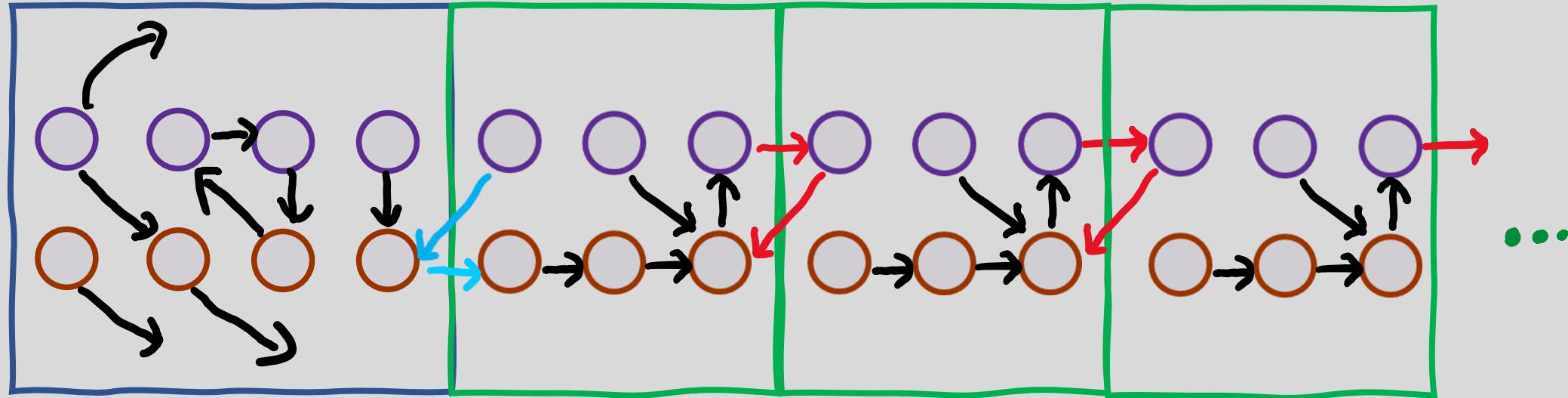
Semilinear
Strategies

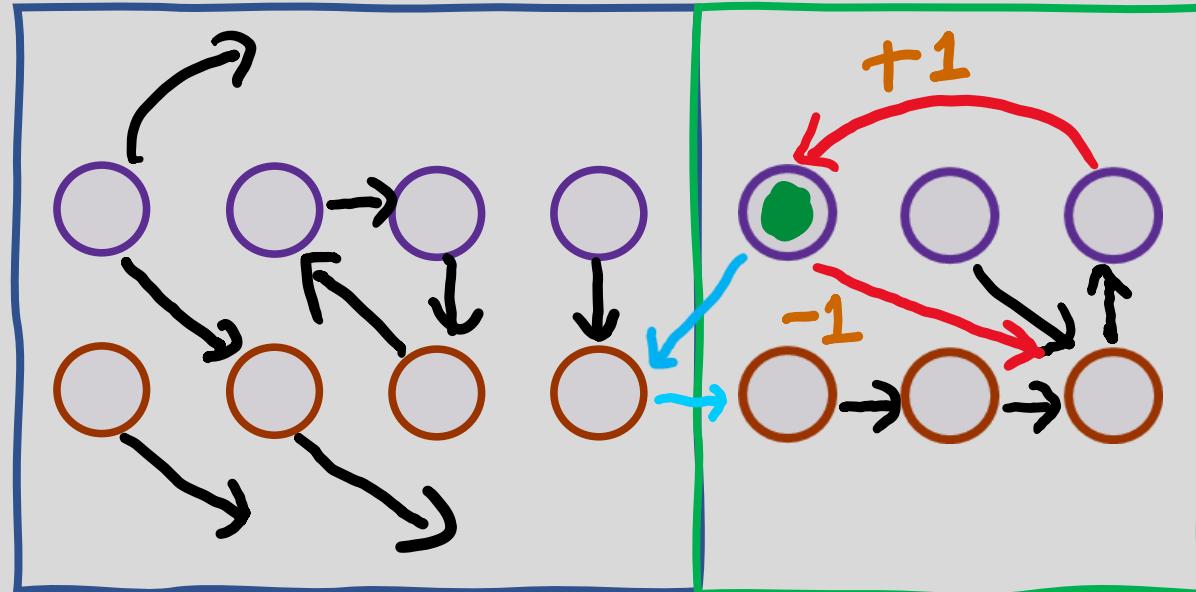
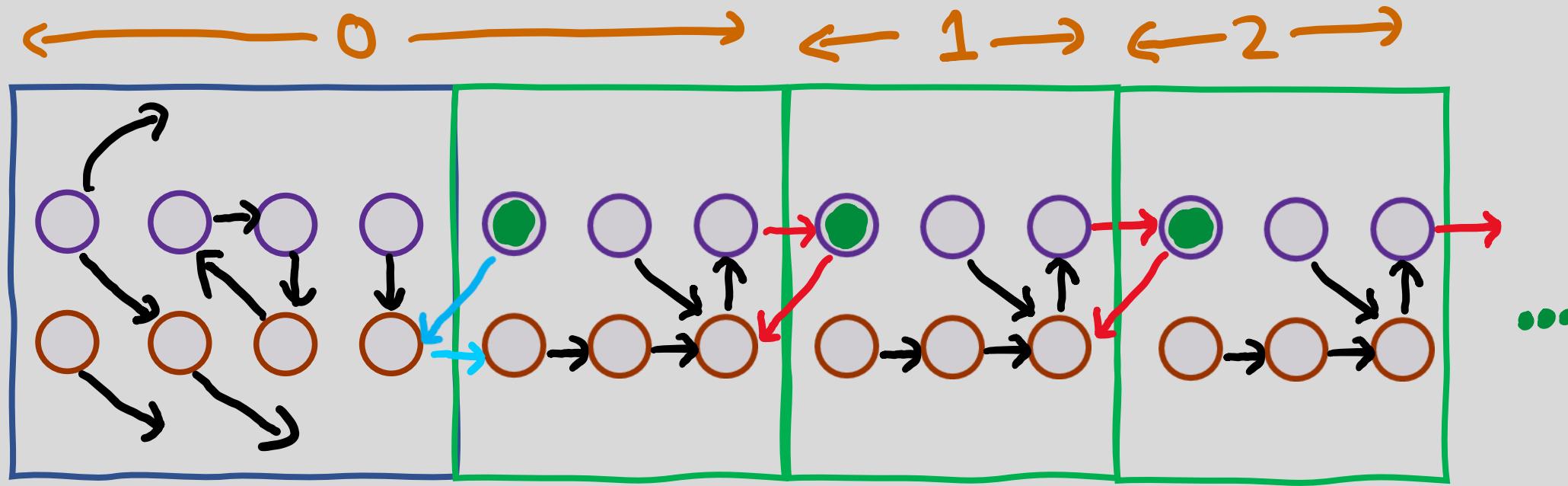
Theorem: Given an history-deterministic OCN,
there is a semilinear strategy for  that
is effectively computable.

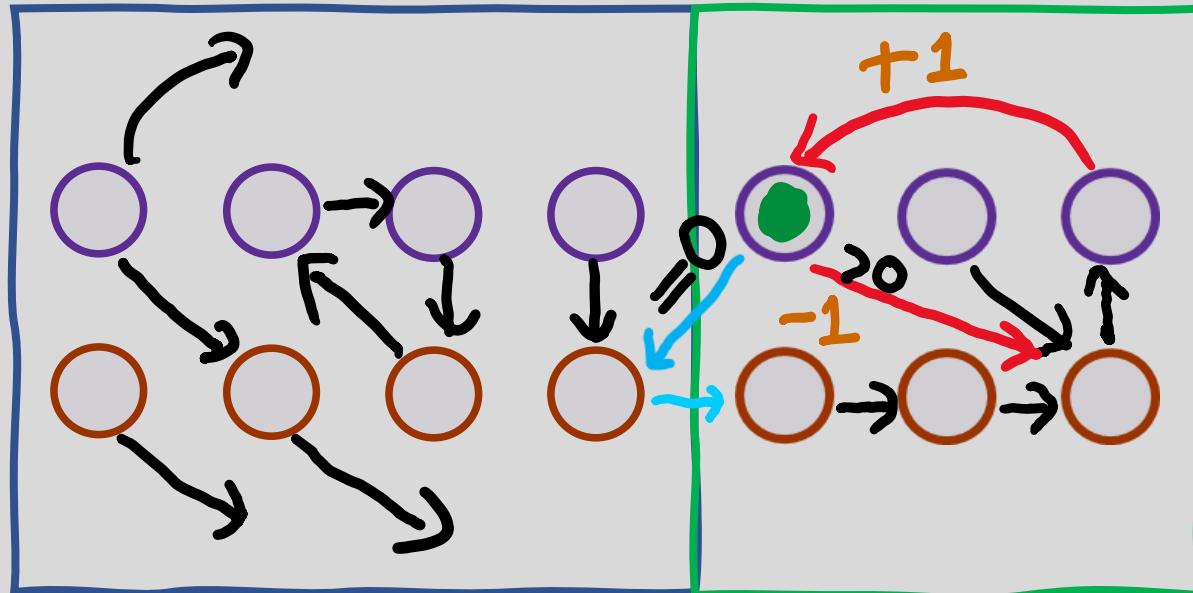
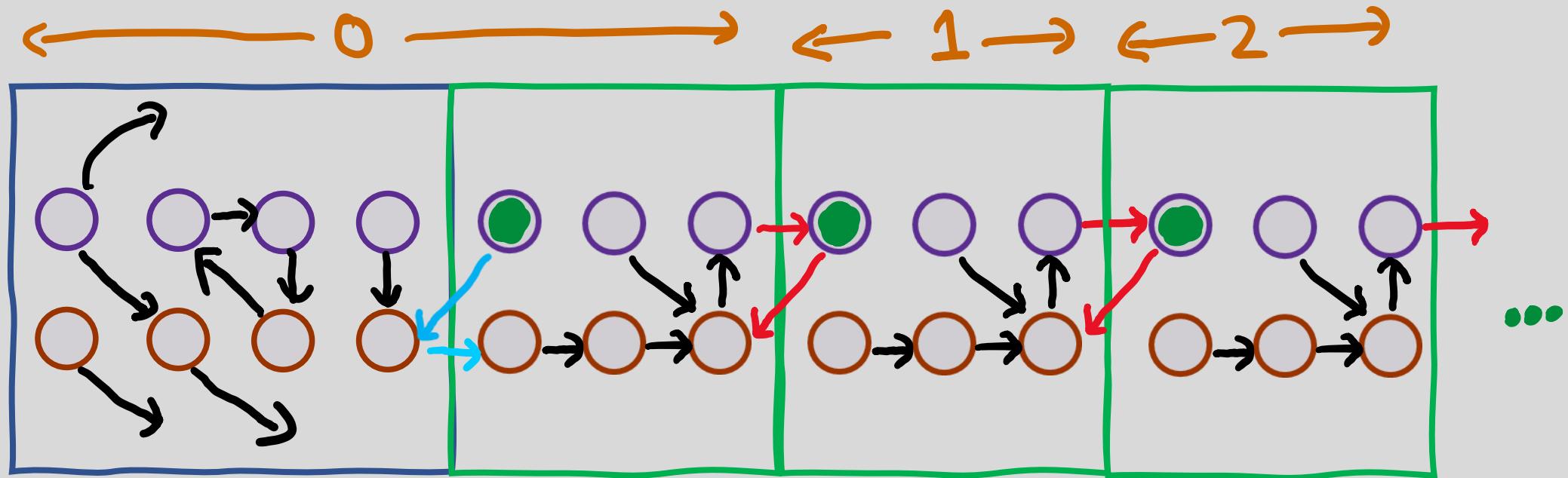




Note that any run that follows 's winning strategy is accepting on all accepting words.







Deterministic
One-counter automaton

Theorem: Given an history-deterministic OCN,
there is a semilinear strategy for  that
is effectively computable.

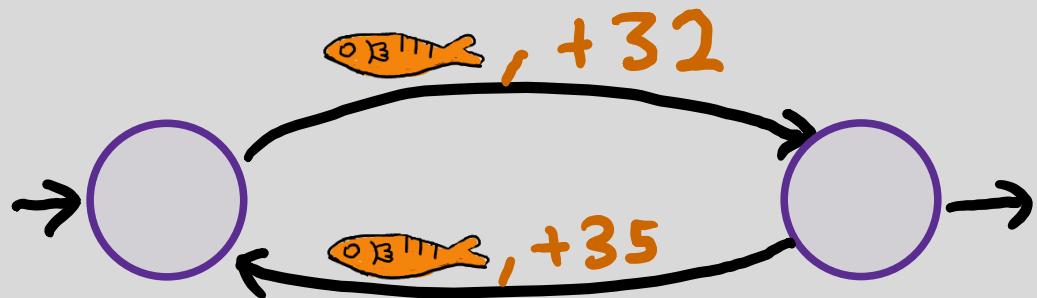
Corollary: Every history-deterministic OCN
can be converted to a language equivalent
deterministic OCA.

Theorem [Hofman, Lasota, Mayr, Totzke '16]

Deciding the winner in the simulation game between two OCNs can be done in PSPACE.

Theorem: Deciding whether a given OCN
is history-deterministic is PSPACE-complete,
and EXPSPACE-complete when the counters
are encoded in binary.

Theorem: Deciding whether a given OCN
is history-deterministic is PSPACE-complete,
and EXPSPACE-complete when the counters
are encoded in binary.



Theorem: Deciding whether a given OCN
is history-deterministic is PSPACE-complete,
and EXPSPACE-complete when the counters
are encoded in binary.

Proof: Upper bound - Reduction to Simulation game

Theorem: Deciding whether a given OCN
is history-deterministic is PSPACE-complete,
and EXPSPACE-complete when the counters
are encoded in binary.

Proof: Upper bound - Reduction to Simulation game

Lower bound - Reduction from Reachability
game on OCN

History-Deterministic One-Counter Nets

- The resolvers in an HD-OCN are semilinear
Corollary: Every HD-OCN can be converted to a deterministic one-counter automaton.
- Complexity of checking history-determinism:
 1. PSPACE-complete for unary encoding
 2. EXPSPACE-complete for binary encoding

3. History - Deterministic

One-Counter Automata

What about One-Counter Automata?

What about One-Counter Automata?



Checking for history-determinism:

What about One-Counter Automata?



Checking for history-determinism: Undecidable

What about One-Counter Automata?



Checking for history-determinism: Undecidable



Do all HD-OCA admit language-equivalent
DOCA?

What about One-Counter Automata?



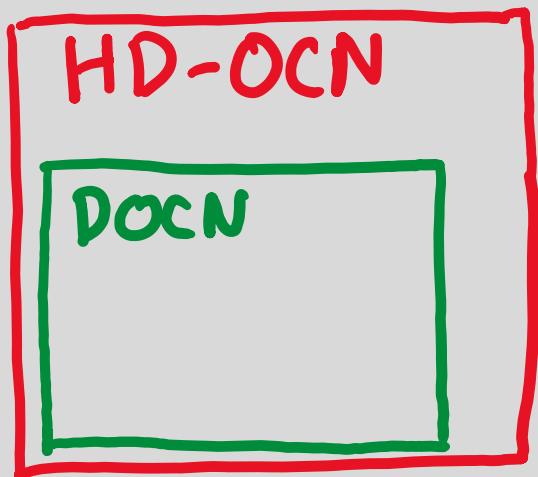
Checking for history-determinism: Undecidable

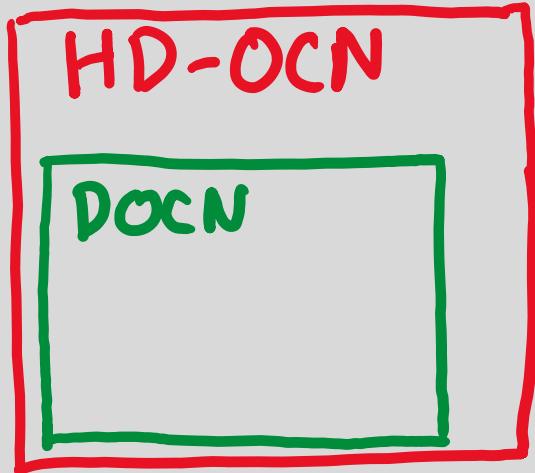


Do all HD-OCA admit language-equivalent
DOCA?

Open. Semilinearity of strategies suffice.



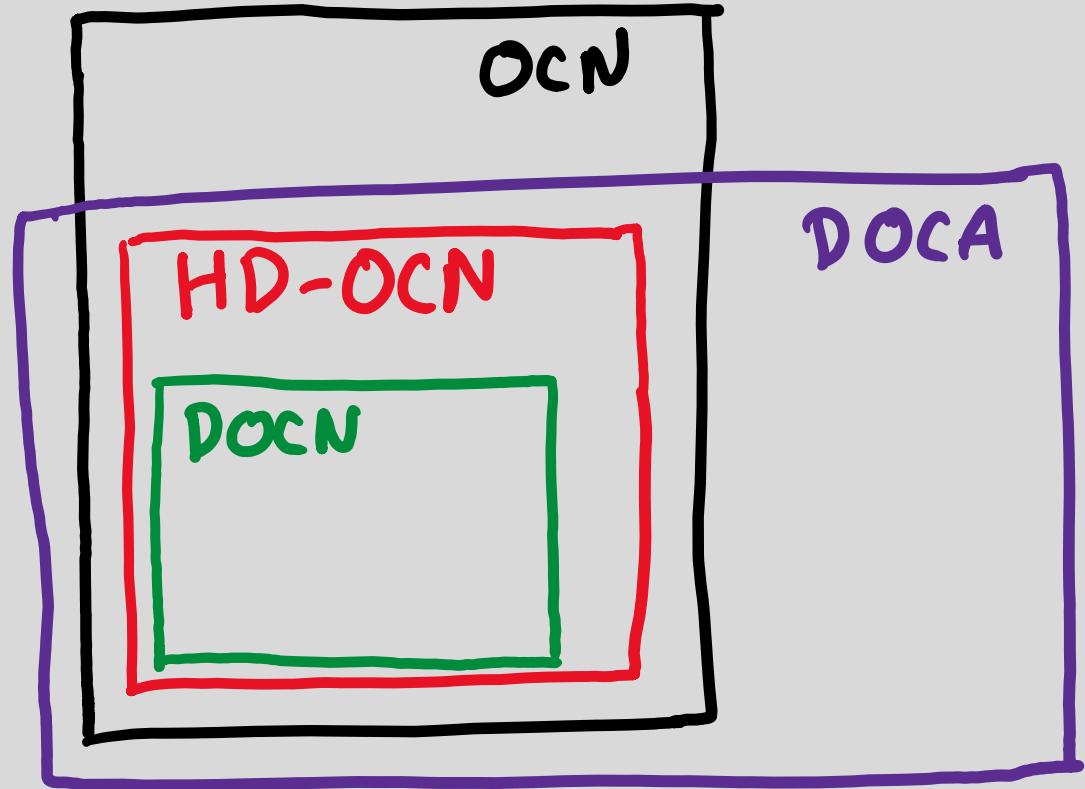




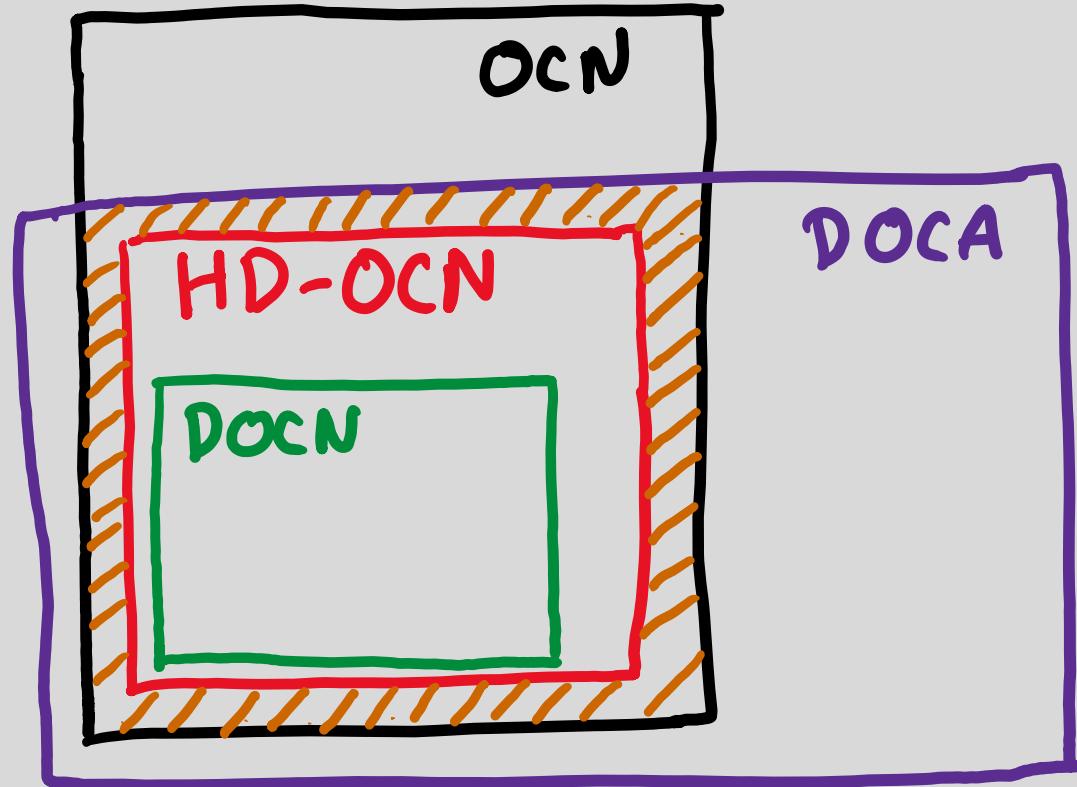
HD-OCNs are more
expressive than DOCNs.

[S. Almagor, A. Yeshurun]

Private Communication



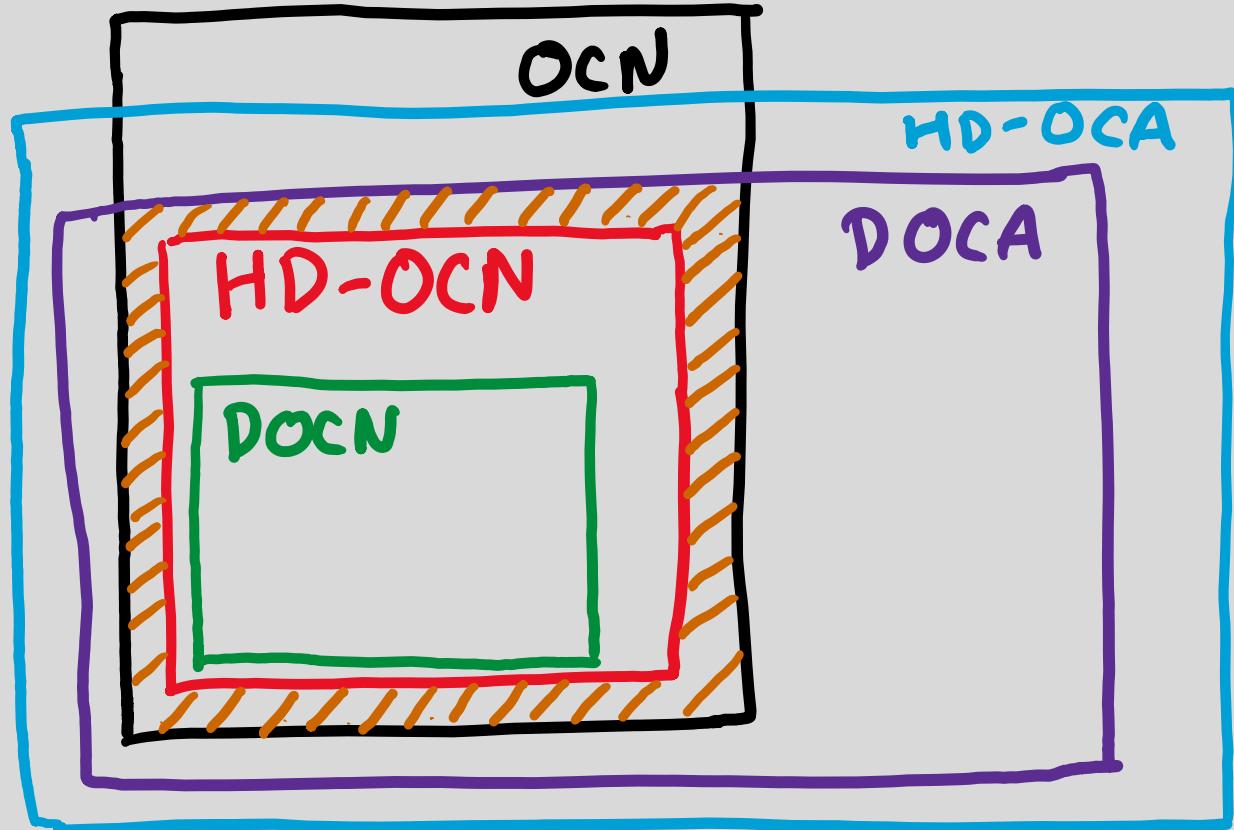
$$HD-OCN \subseteq OCN \cap DOCA$$



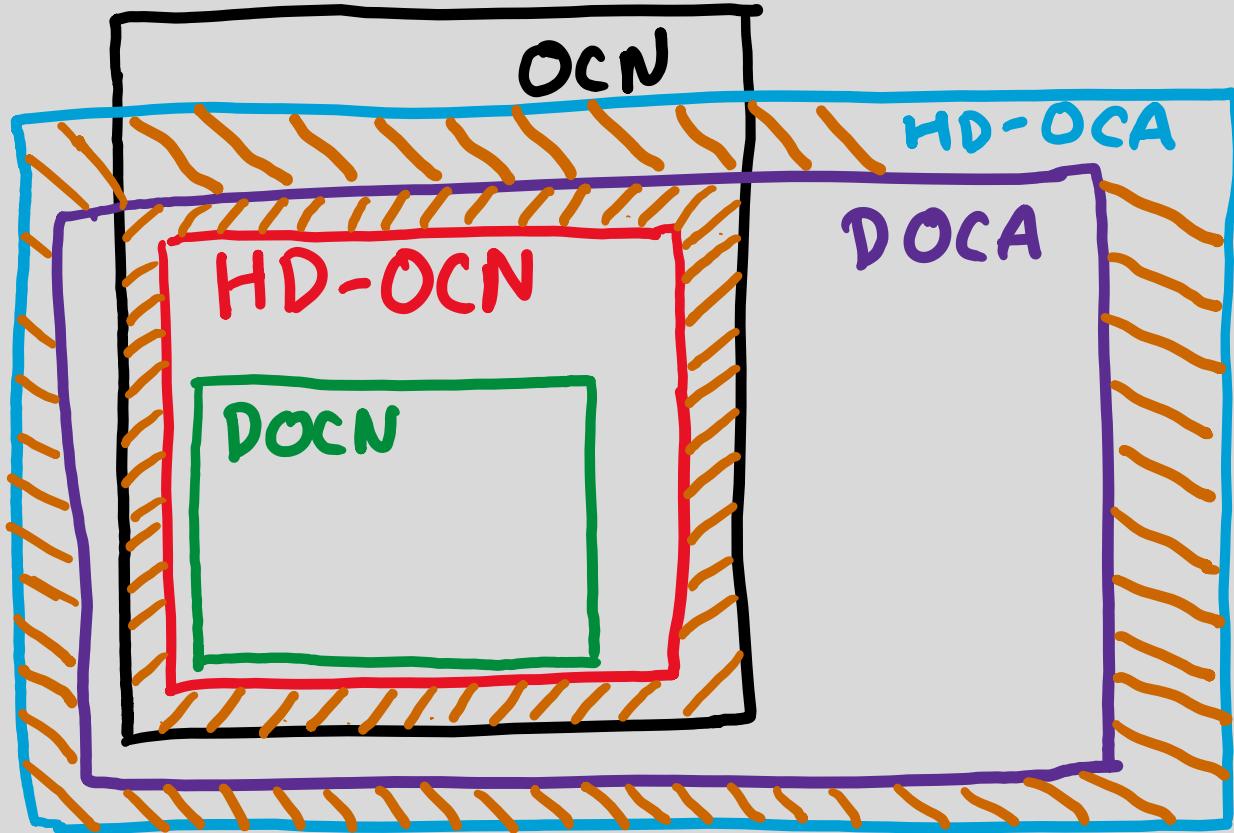
$$HD-OCN \subseteq OCN \cap DOCA$$

Is $HD-OCN = OCN \cap DOCA$?

Open.



Is HD-OCA = DOCA ?



Is $HD-OCA = DOCA$?

Open

Conclusion

-  Every history-deterministic OCN has a language-equivalent DOCA.
-  Checking history-determinism for OCNs is PSPACE-complete.
-  Checking history-determinism for OCAs is undecidable.

Open Problems



Is $\text{HD-OCA} = \text{DOCA}$?



Is HD-OCN
 $= \text{OCN} \cap \text{DOCA}$?

