Robust control of Boolean networks in systems biology

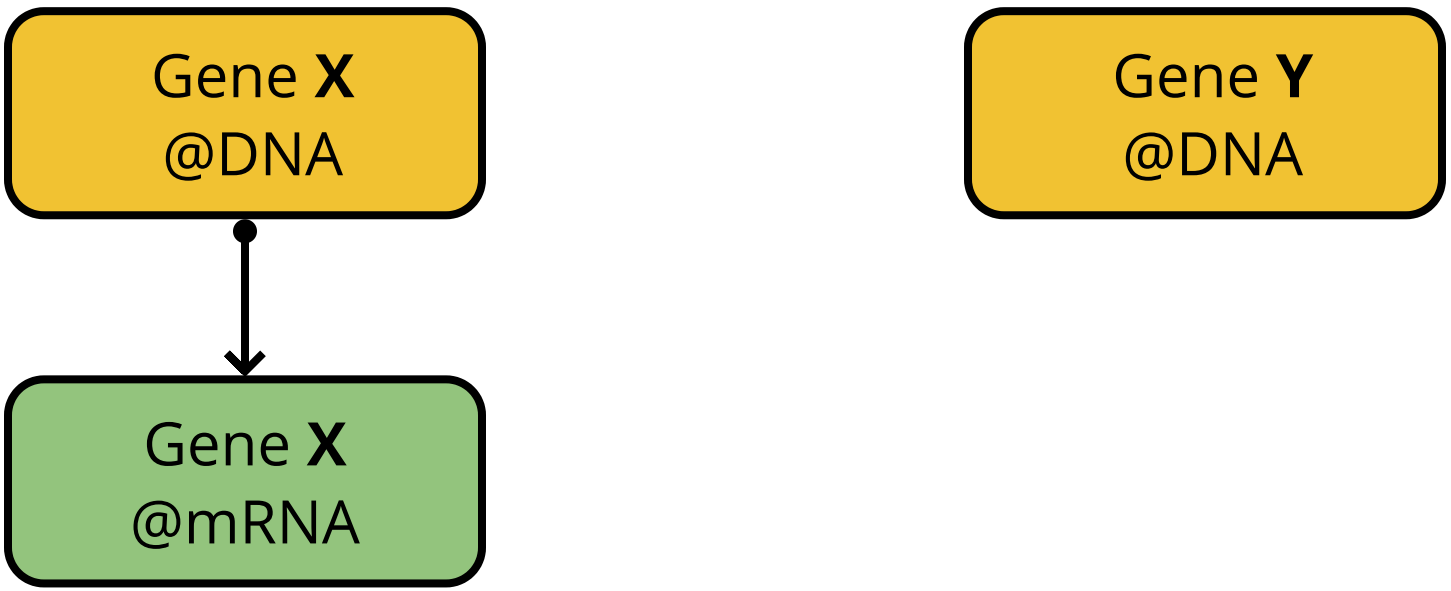
Sam Pastva,
with Eva Šmijáková, David Šafránek, and Luboš Brim

Modelling gene interactions

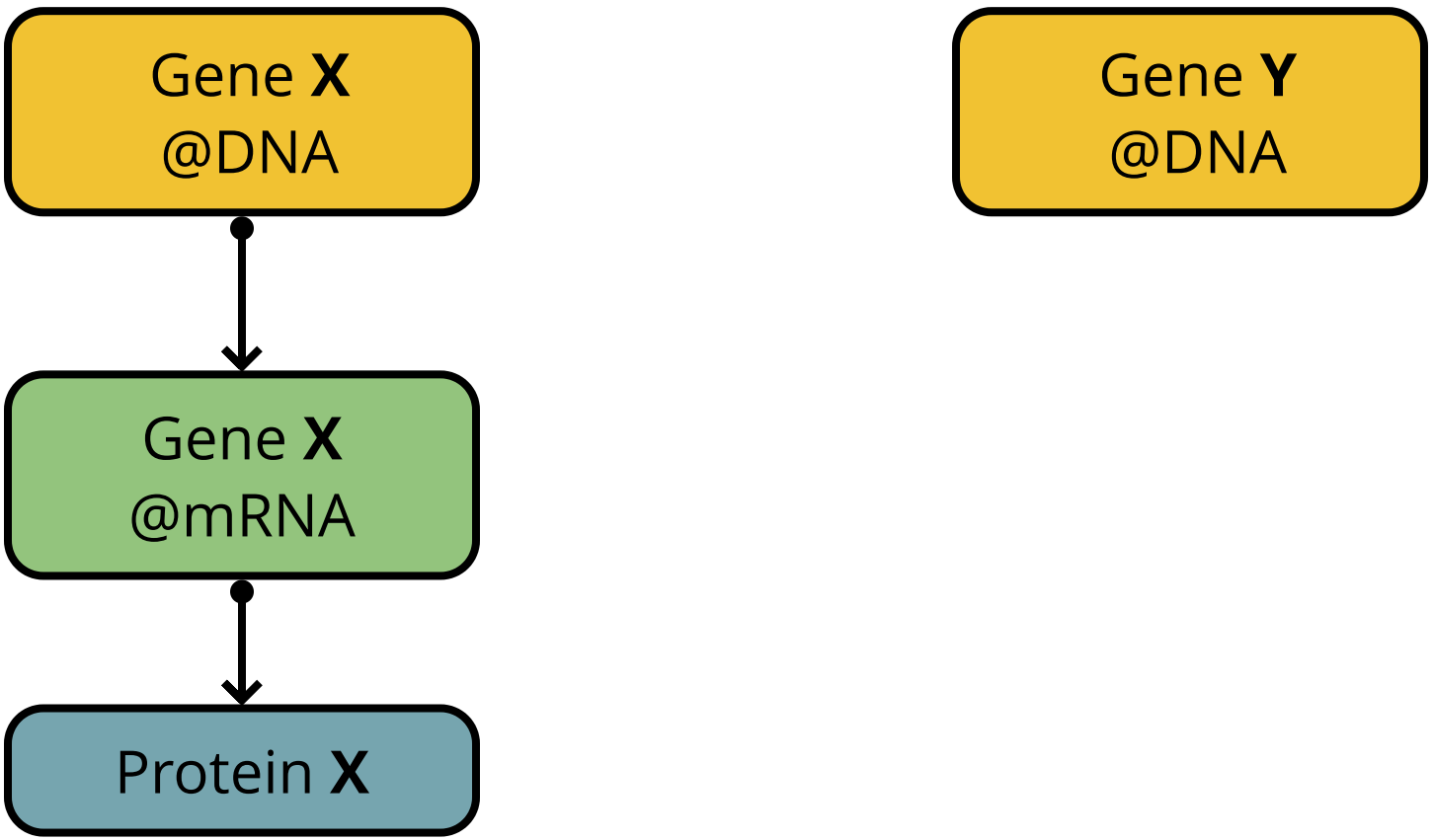
Modelling gene interactions

Modelling gene interactions

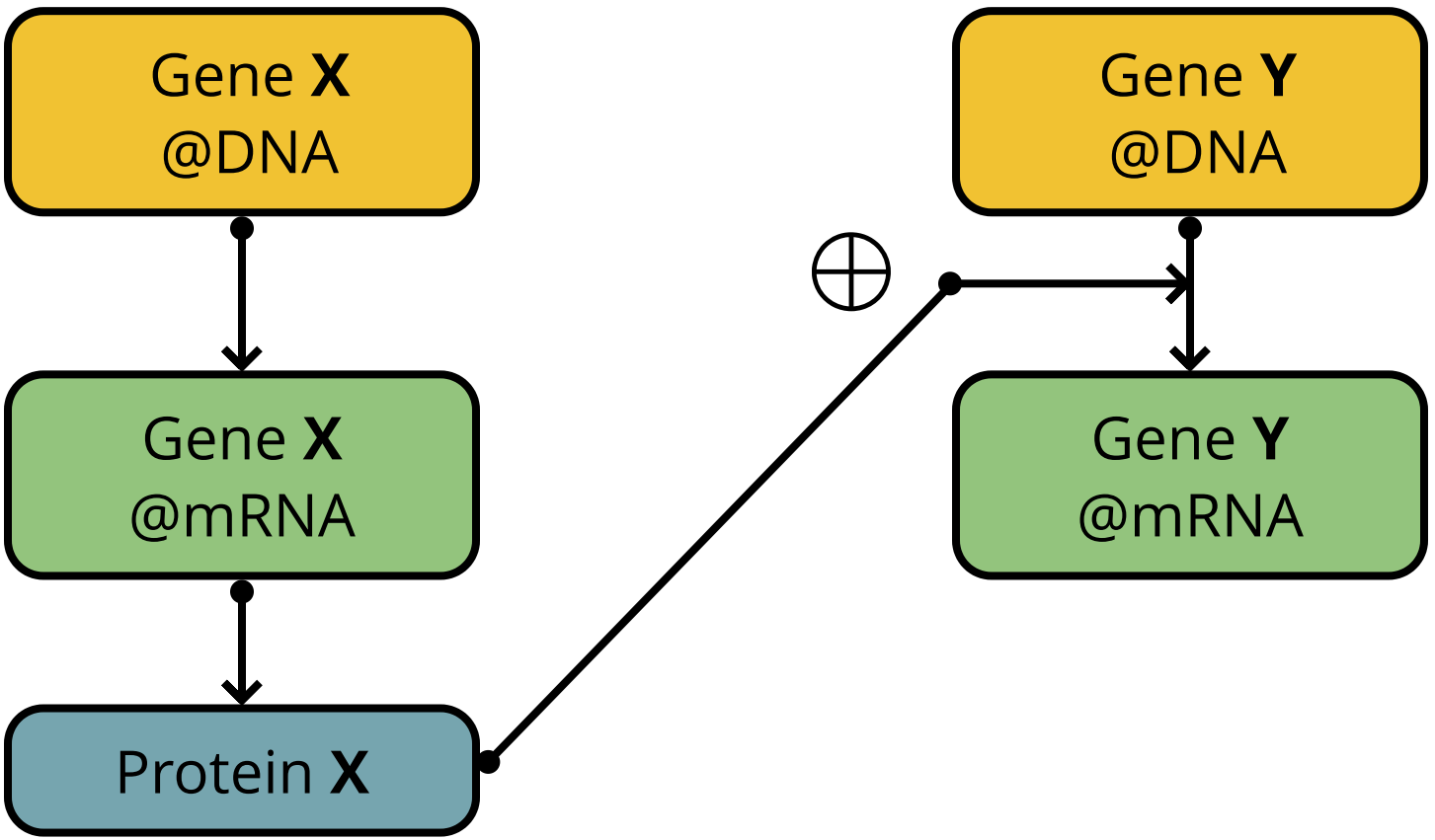
Modelling gene interactions



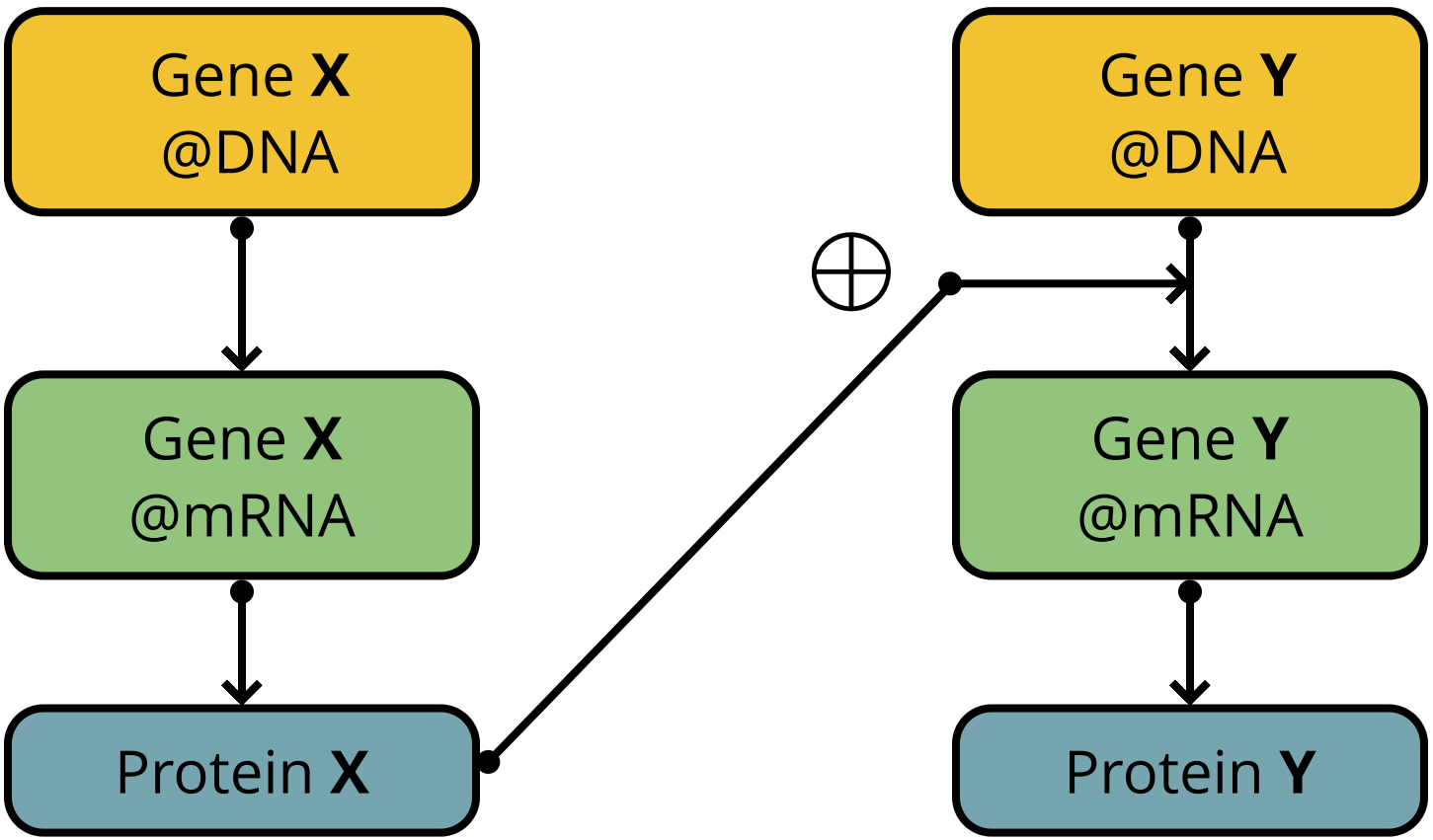
Modelling gene interactions



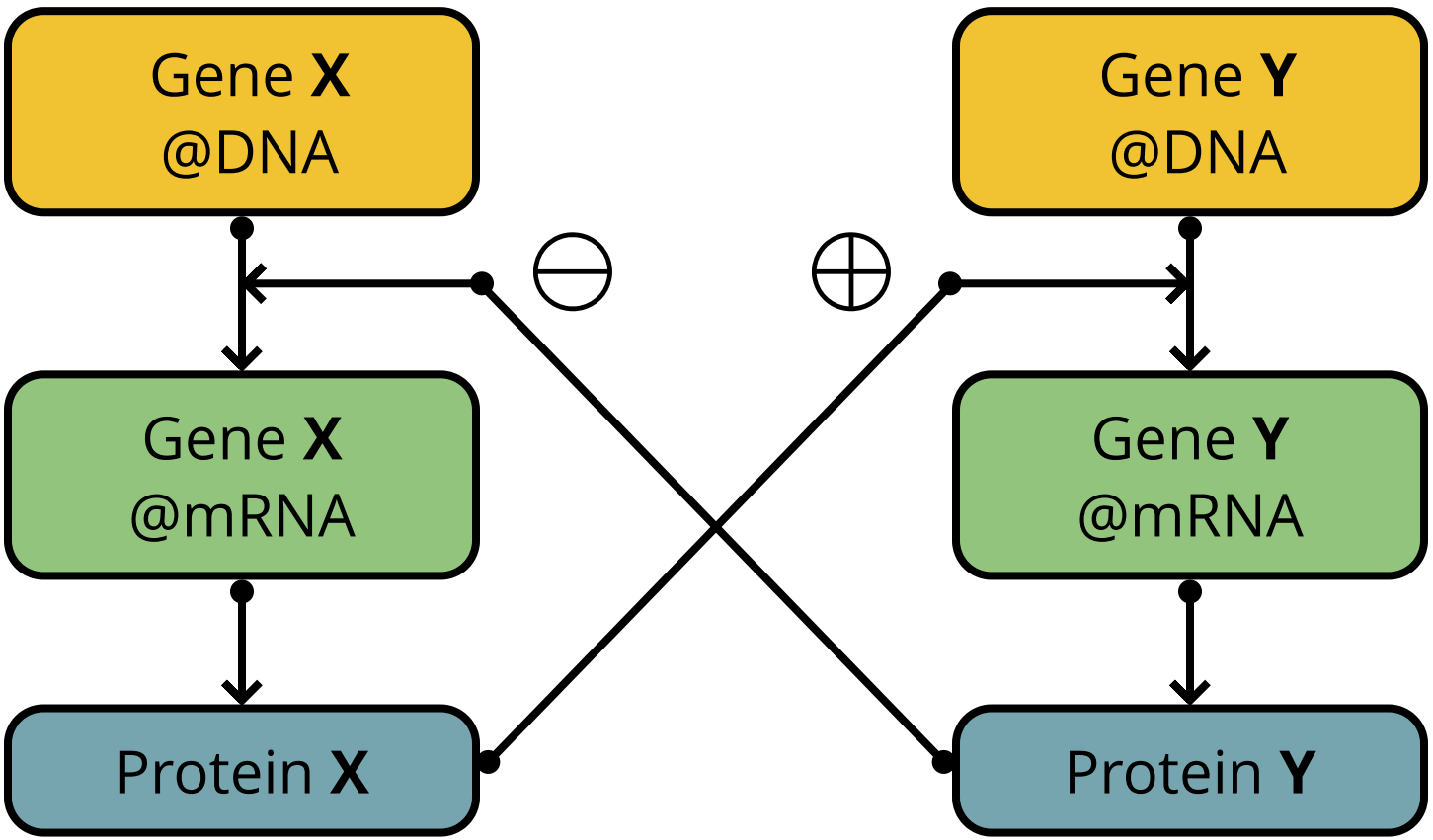
Modelling gene interactions



Modelling gene interactions

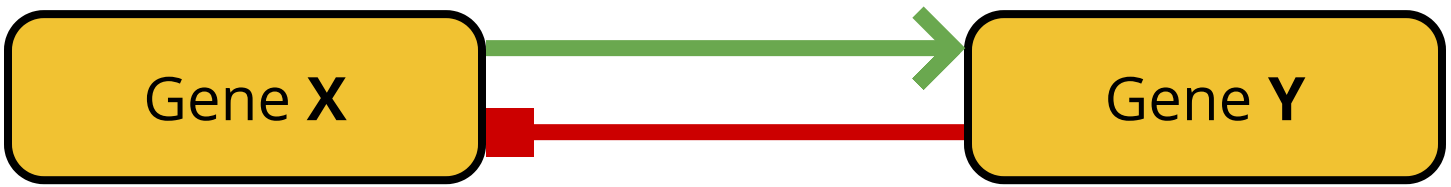


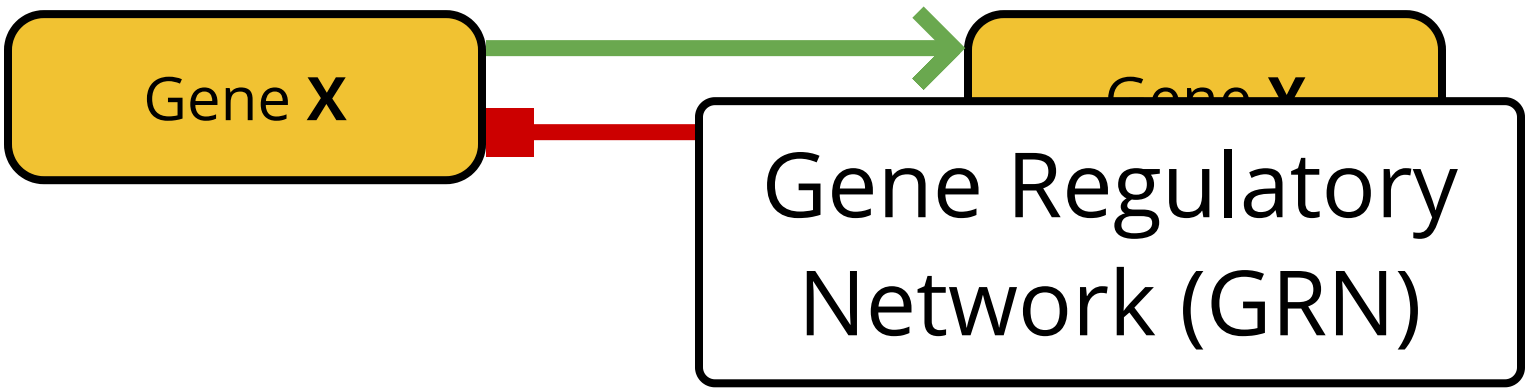
Modelling gene interactions

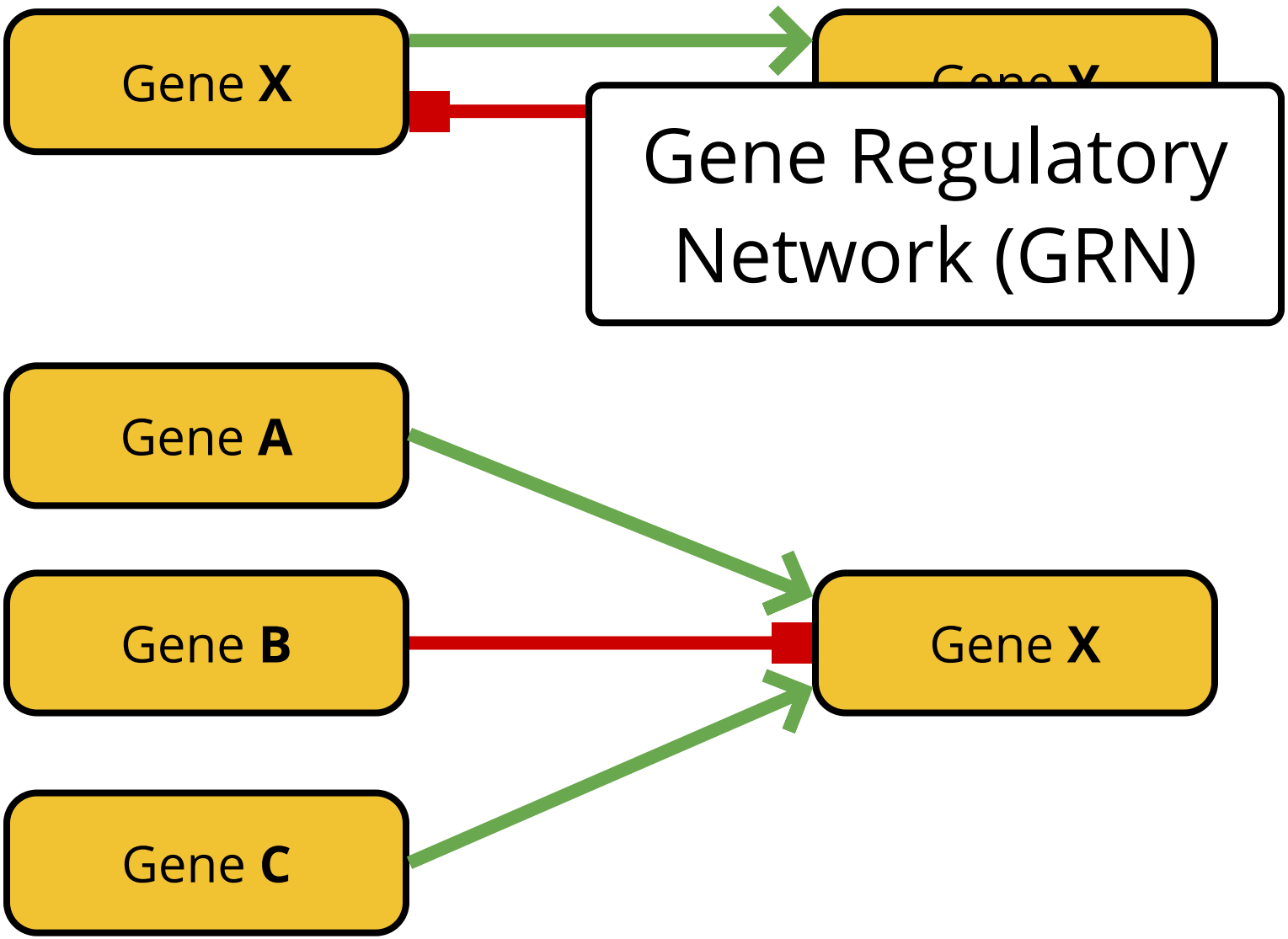


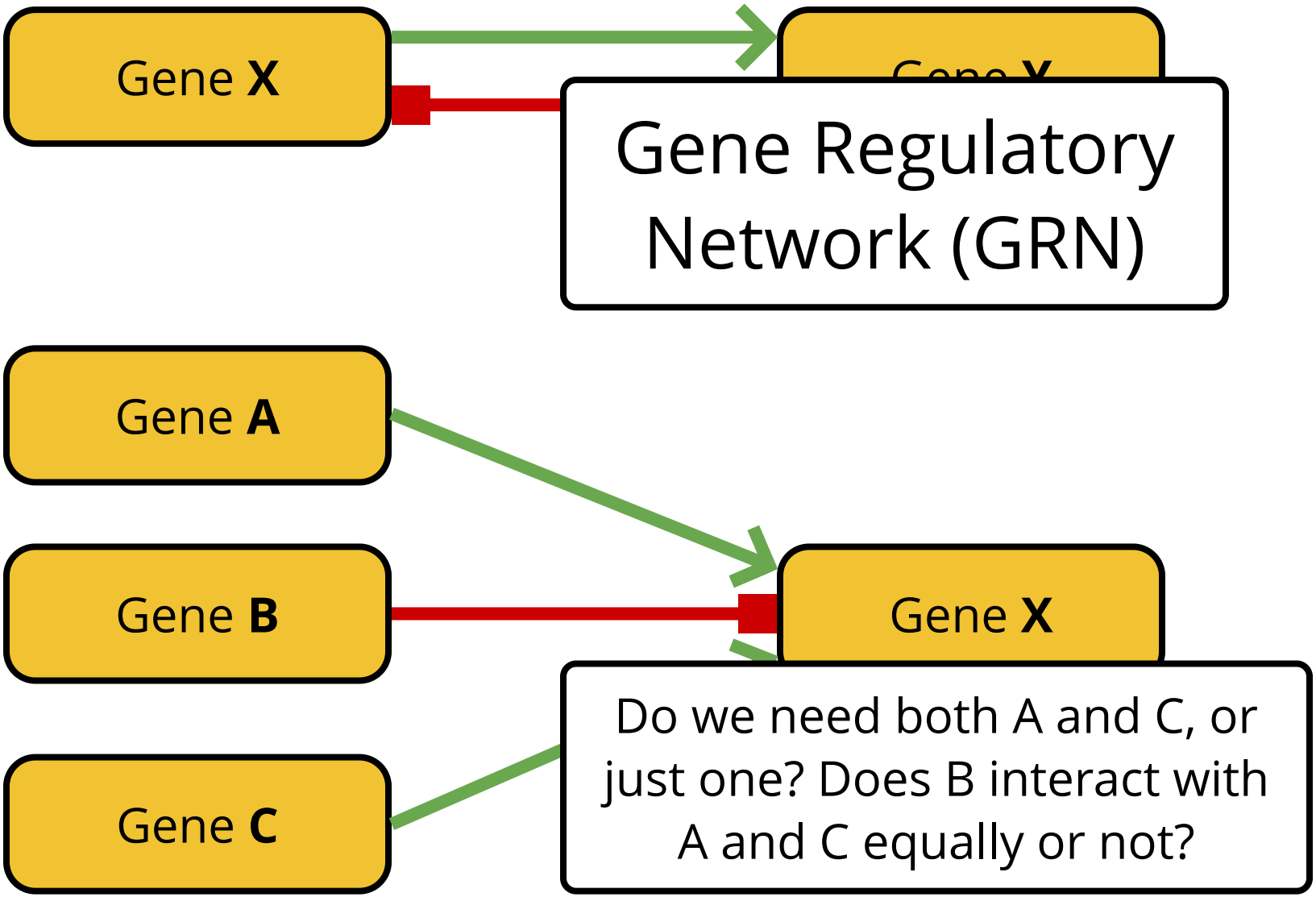

Gene **X**
Gene **Y**

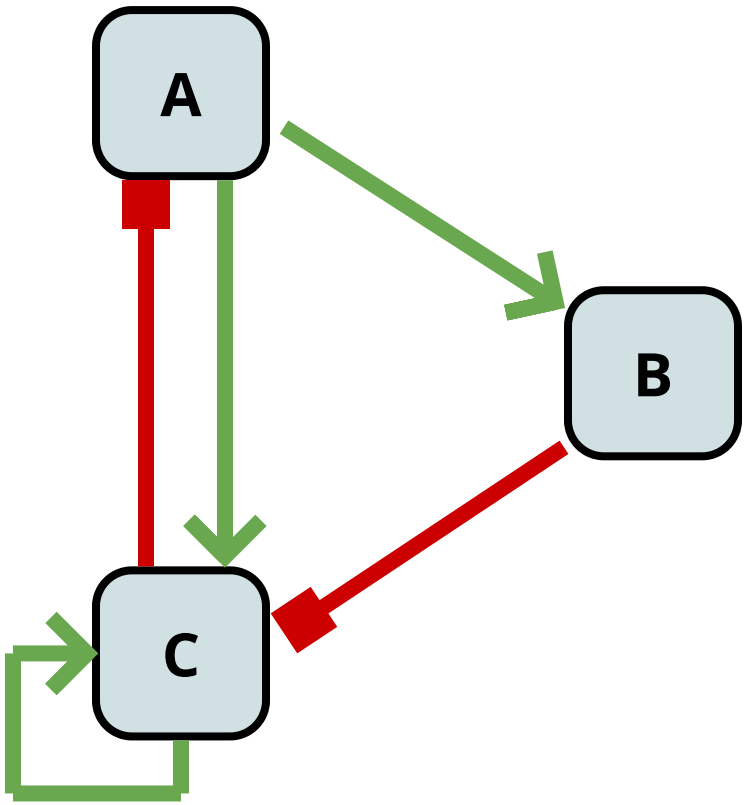

Gene **X**
Gene **Y**

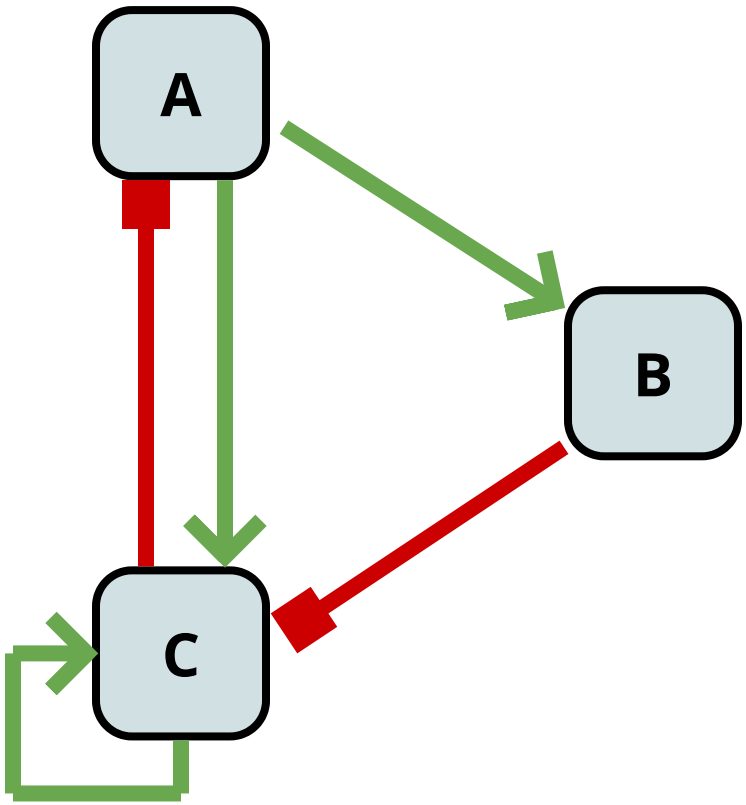

Gene X
Gene Y


Gene **X**
Gene Regulatory Network (GRN)

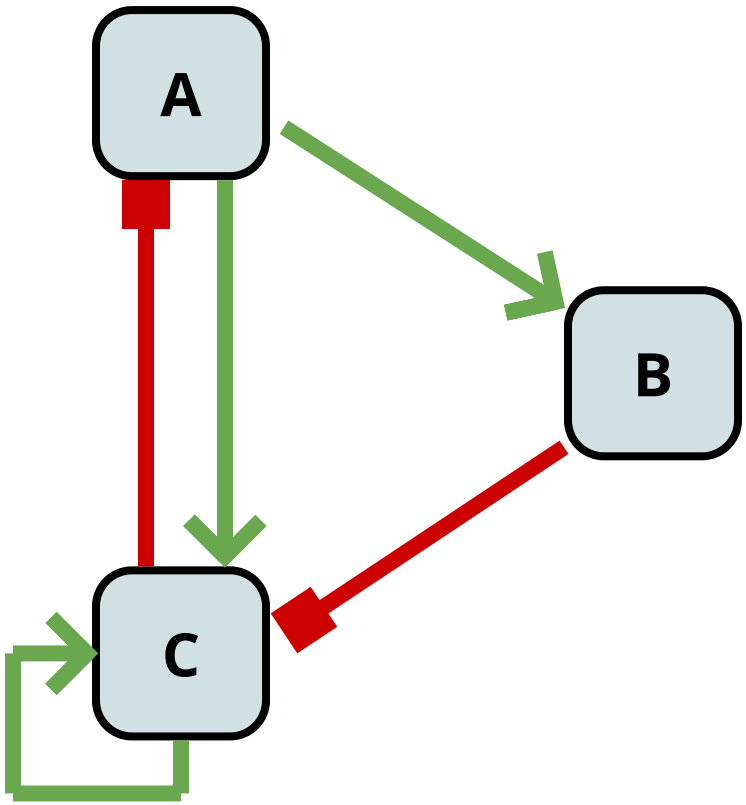

Gene X
Gene Regulatory Network (GRN)
Gene A
Gene B
Gene C
Gene X


Gene X
Gene Regulatory Network (GRN)
Gene A
Gene B
Gene C
Gene X
Do we need both A and C, or just one? Does B interact with A and C equally or not?


A
B
C

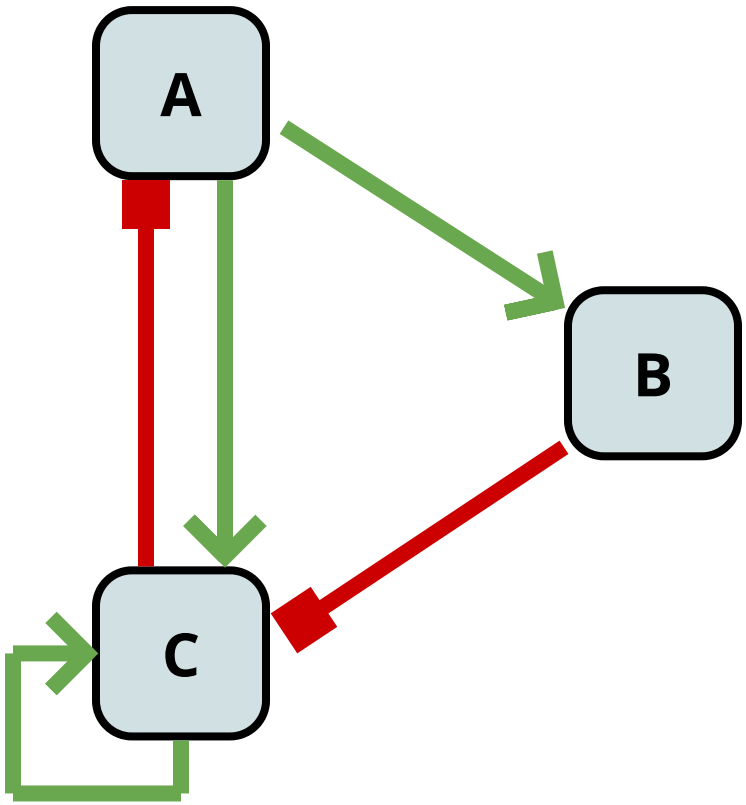


$$x \in \{0,1\}^n \text{ (state)}$$



$$x \in \{0,1\}^n \text{ (state)}$$

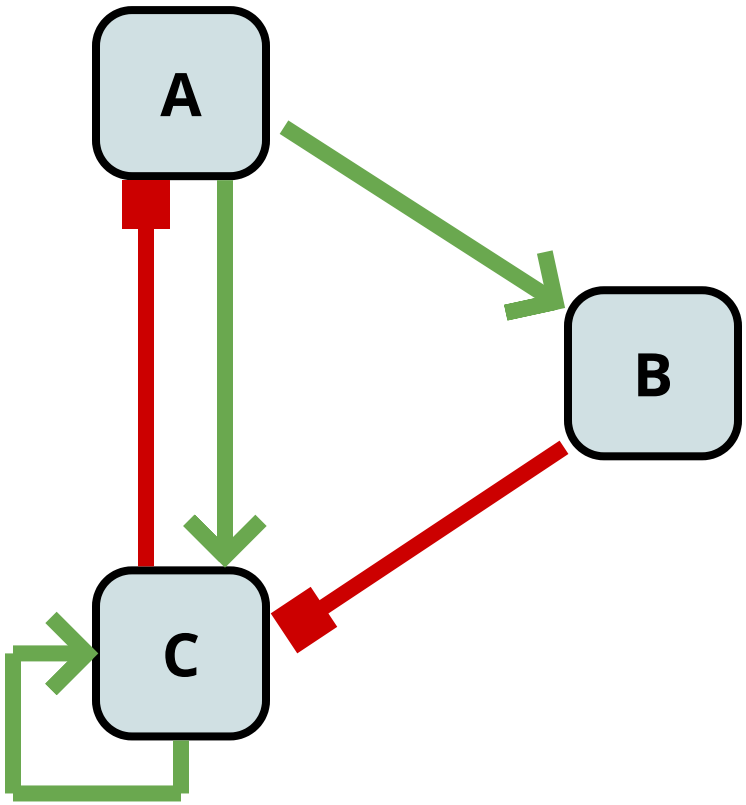
$$f_A(x) = \neg x_C$$



$$x \in \{0,1\}^n \text{ (state)}$$

$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

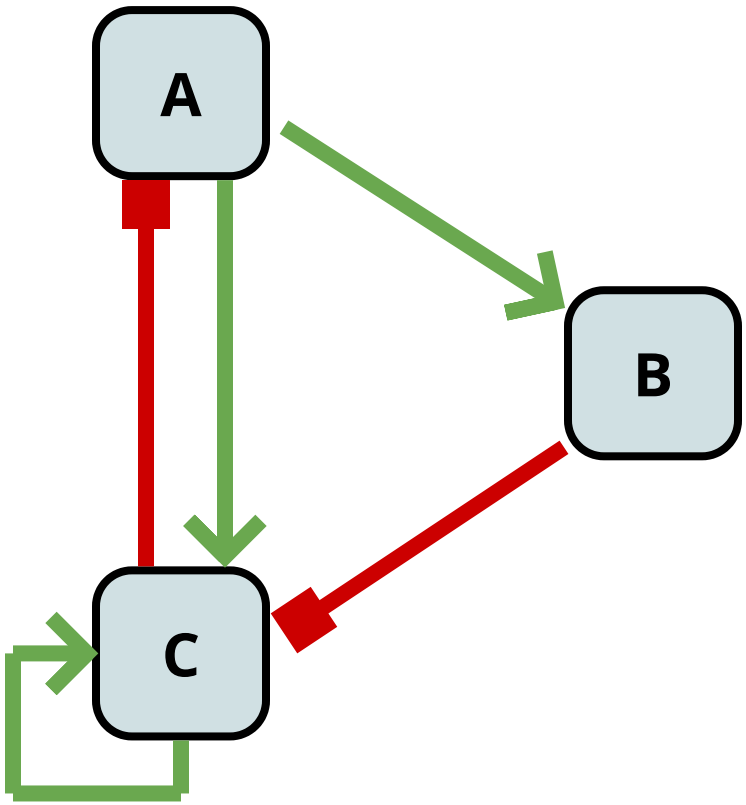


$$x \in \{0,1\}^n \text{ (state)}$$

$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



Boolean Network (BN)

$$x \in \{0,1\}^n \text{ (state)}$$

$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$

$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

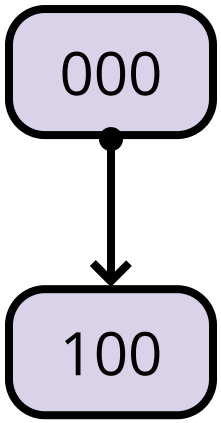
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$

000

$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

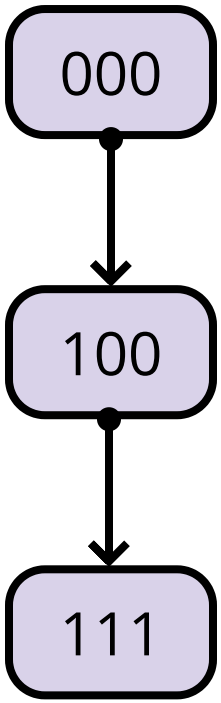
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

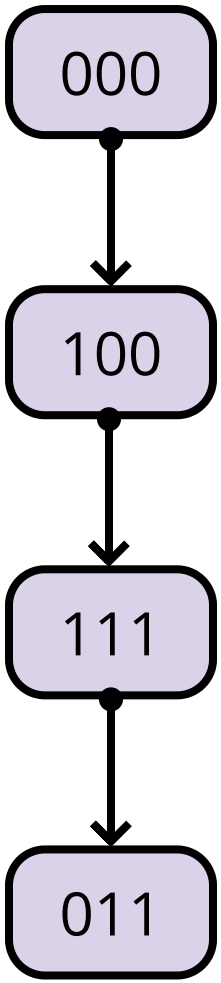
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

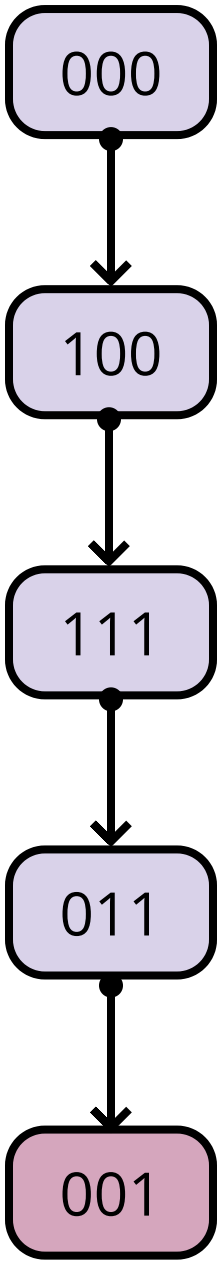
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

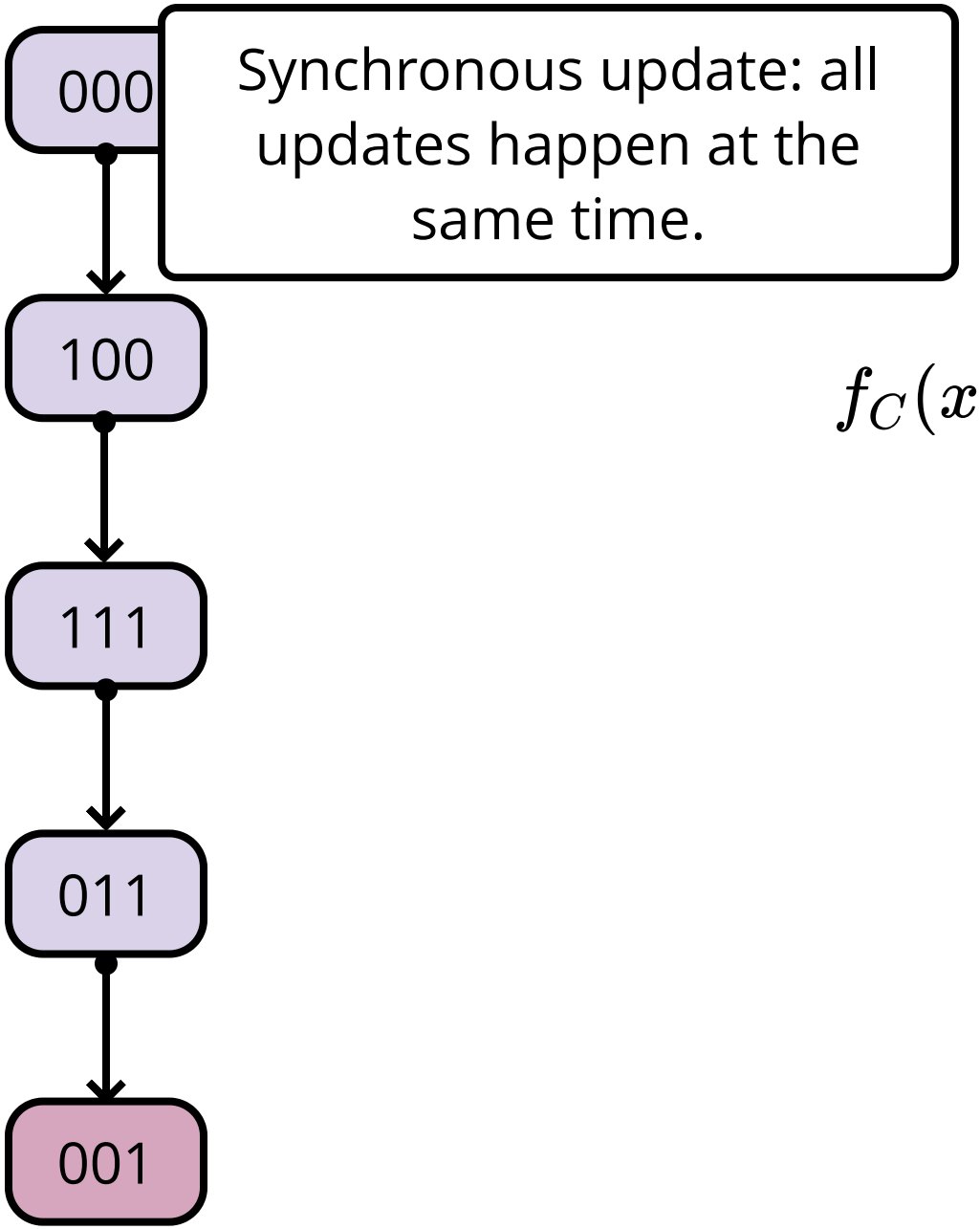
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

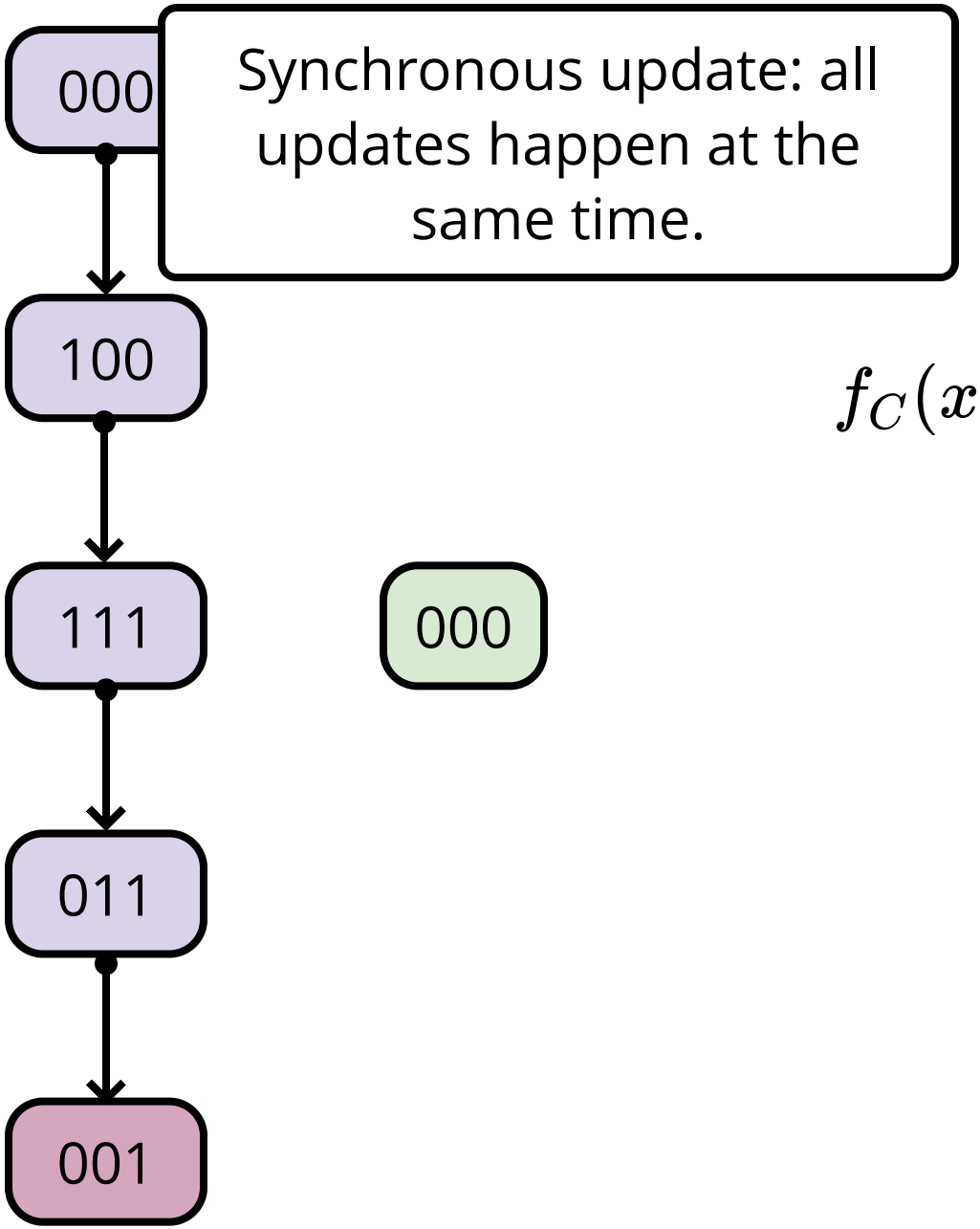
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

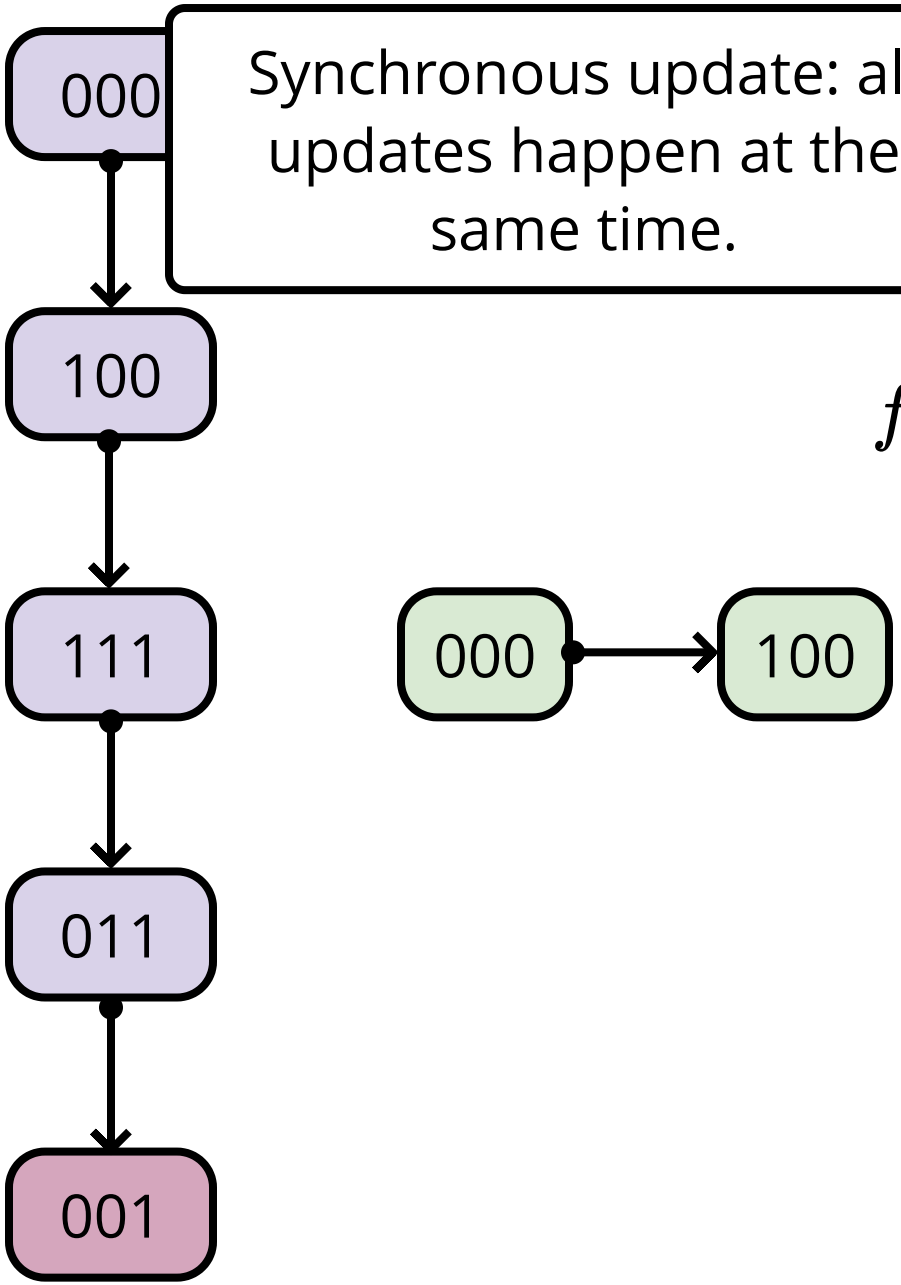
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

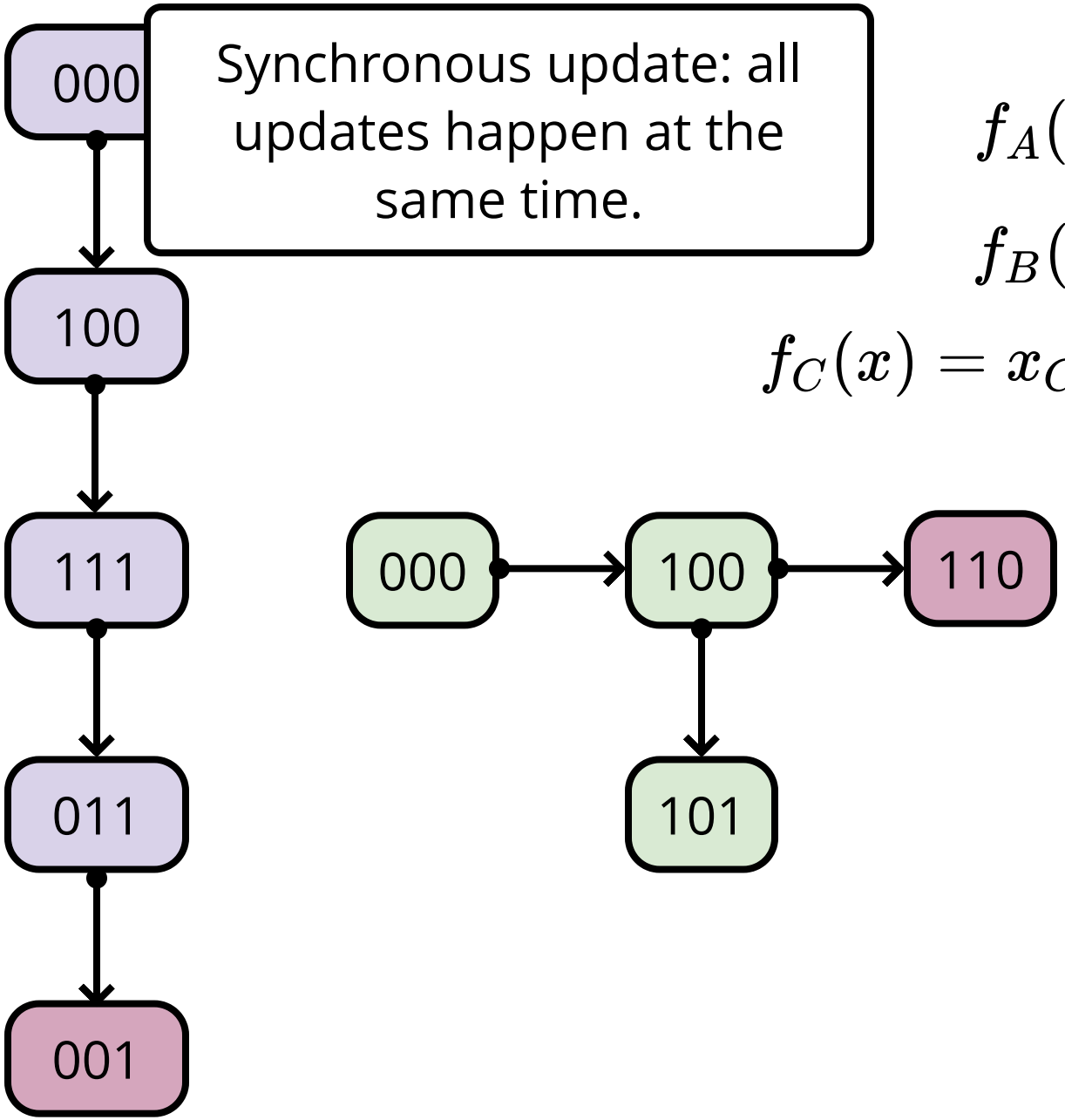
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

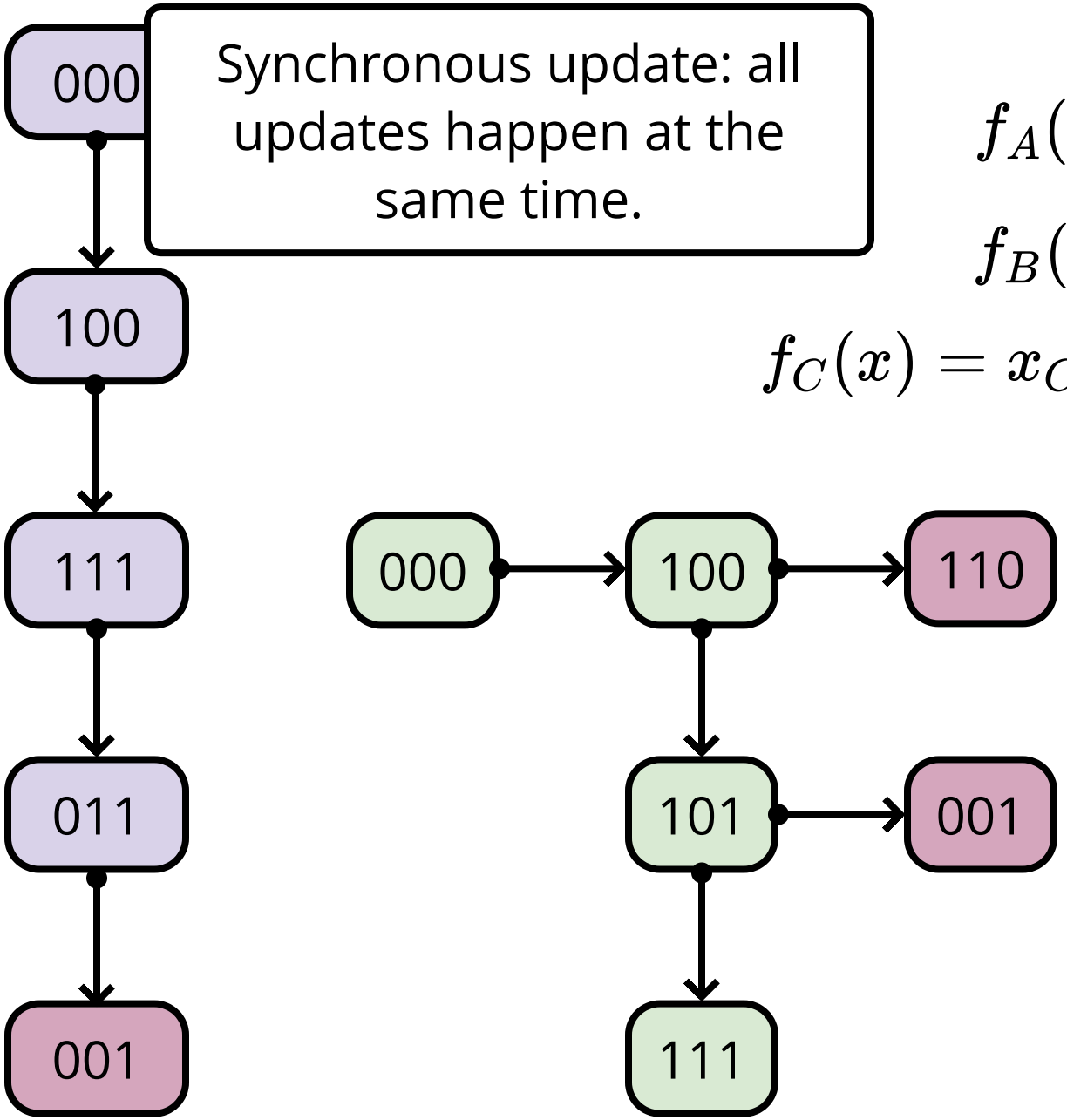
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

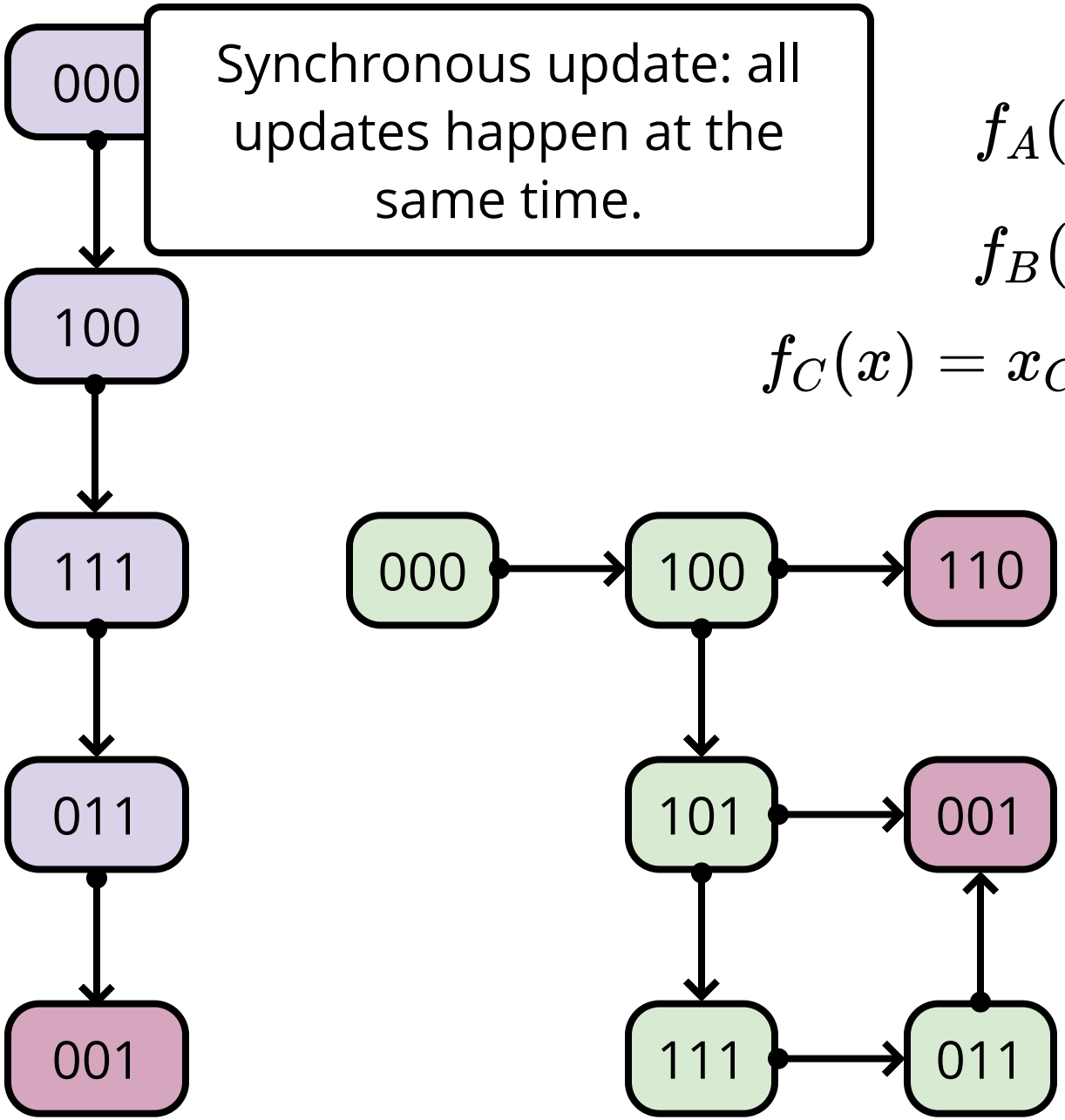
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

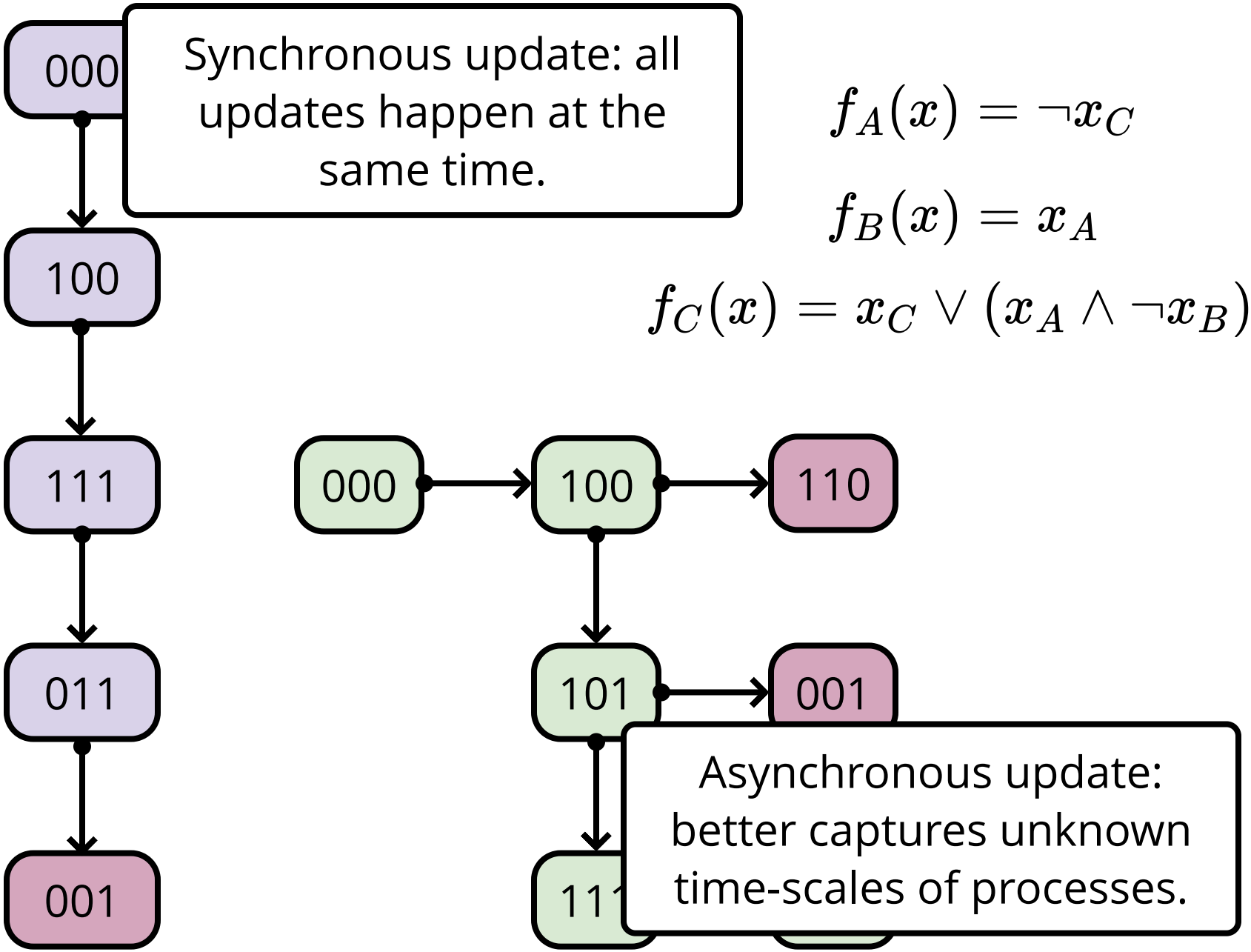
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$



$$f_A(x) = \neg x_C$$

$$f_B(x) = x_A$$

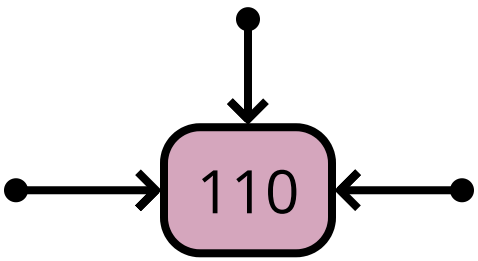
$$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$$


000
100
111
011
001
Synchronous update: all updates happen at the same time.
$f_A(x) = \neg x_C$
$f_B(x) = x_A$
$f_C(x) = x_C \vee (x_A \wedge \neg x_B)$
000
100
110
101
001
Asynchronous update: better captures unknown time-scales of processes.

What "outcomes" are possible in our model? ***(Attractors - bottom SCCs)***

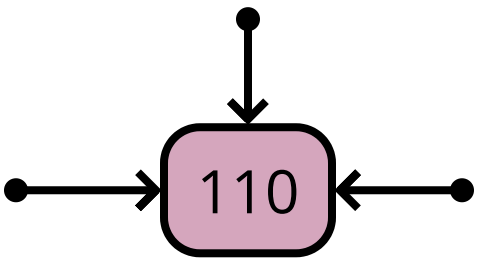
What "outcomes" are possible in our model? ***(Attractors - bottom SCCs)***

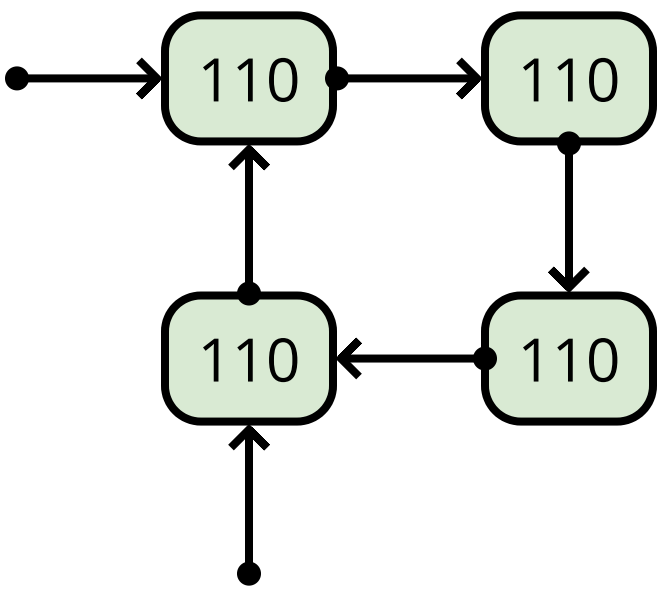
- Single state (*fixed-point attractor*)



What "outcomes" are possible in our model? ***(Attractors - bottom SCCs)***

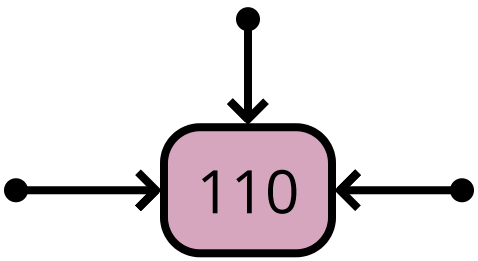
- Single state (*fixed-point attractor*)
- Larger region (*complex attractor*)
  - cycle (*oscillation*)
  - other (*disorder*)

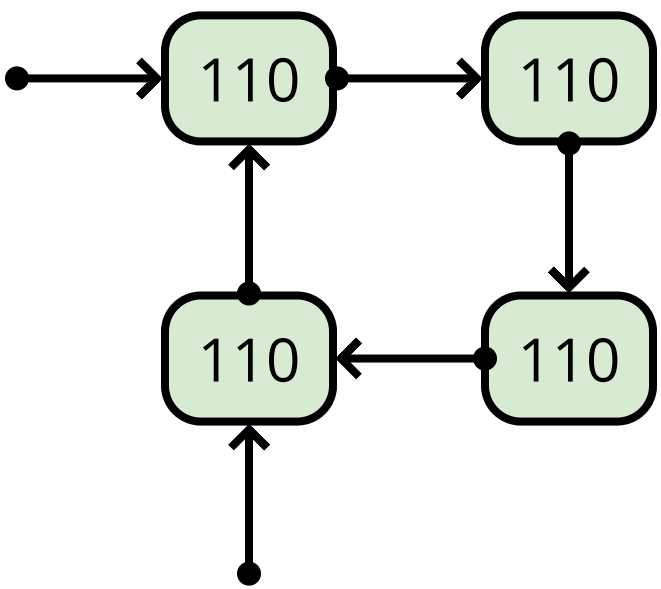


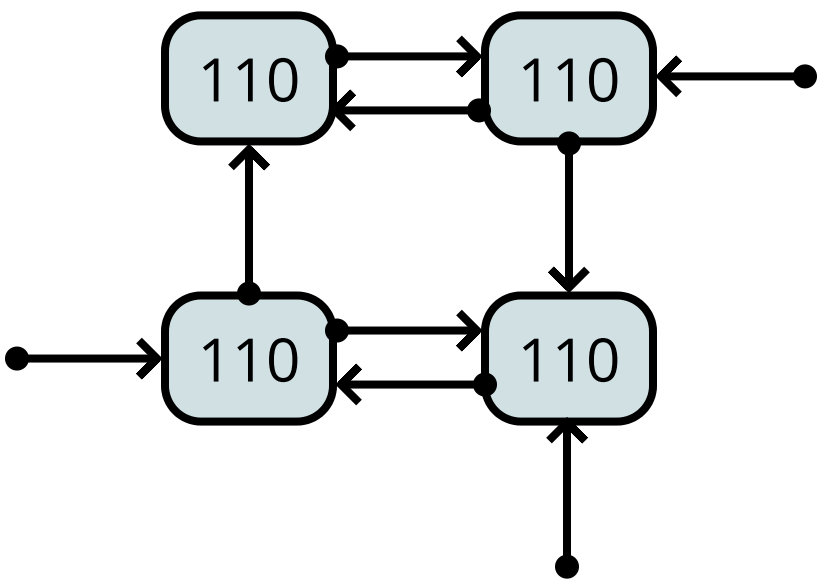


What "outcomes" are possible in our model? ***(Attractors - bottom SCCs)***

- Single state (*fixed-point attractor*)
- Larger region (*complex attractor*)
  - cycle (*oscillation*)
  - other (*disorder*)

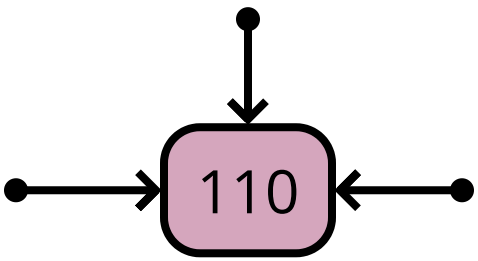


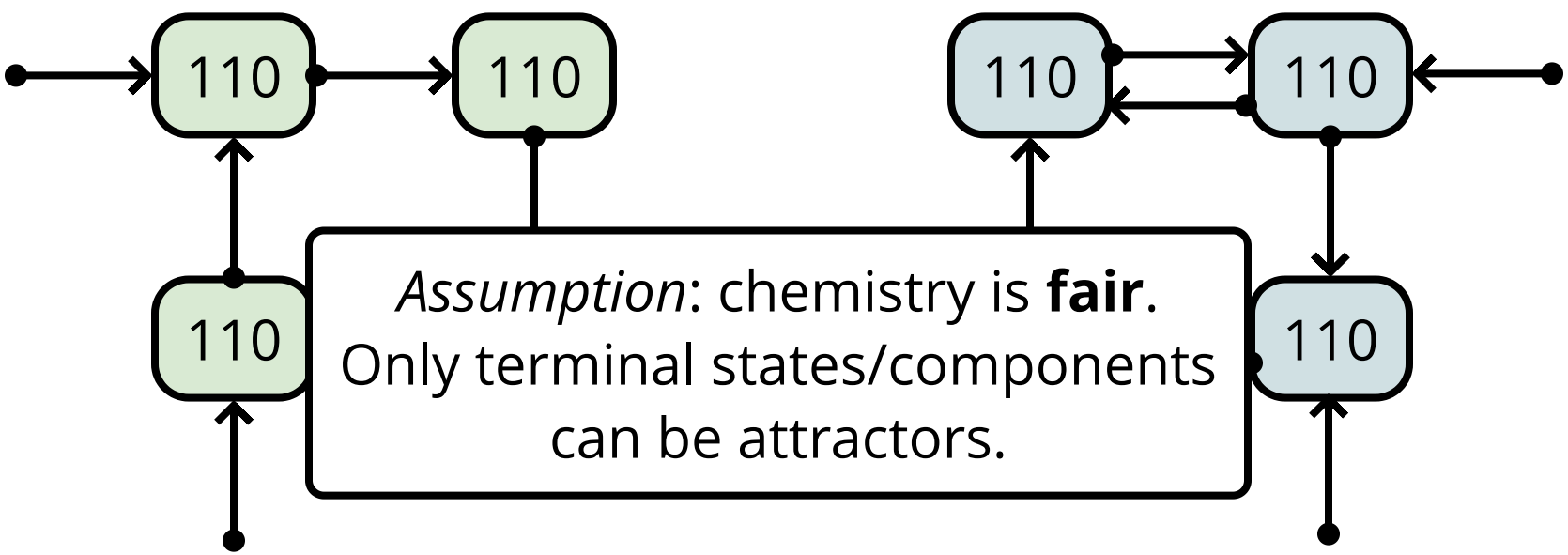




What "outcomes" are possible in our model? ***(Attractors - bottom SCCs)***

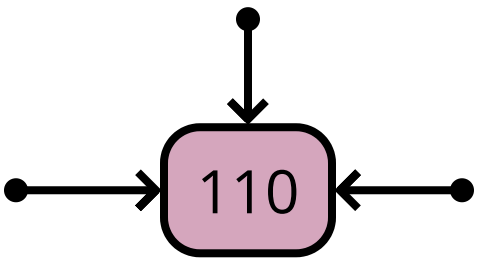
- Single state (*fixed-point attractor*)
- Larger region (*complex attractor*)
  - cycle (*oscillation*)
  - other (*disorder*)

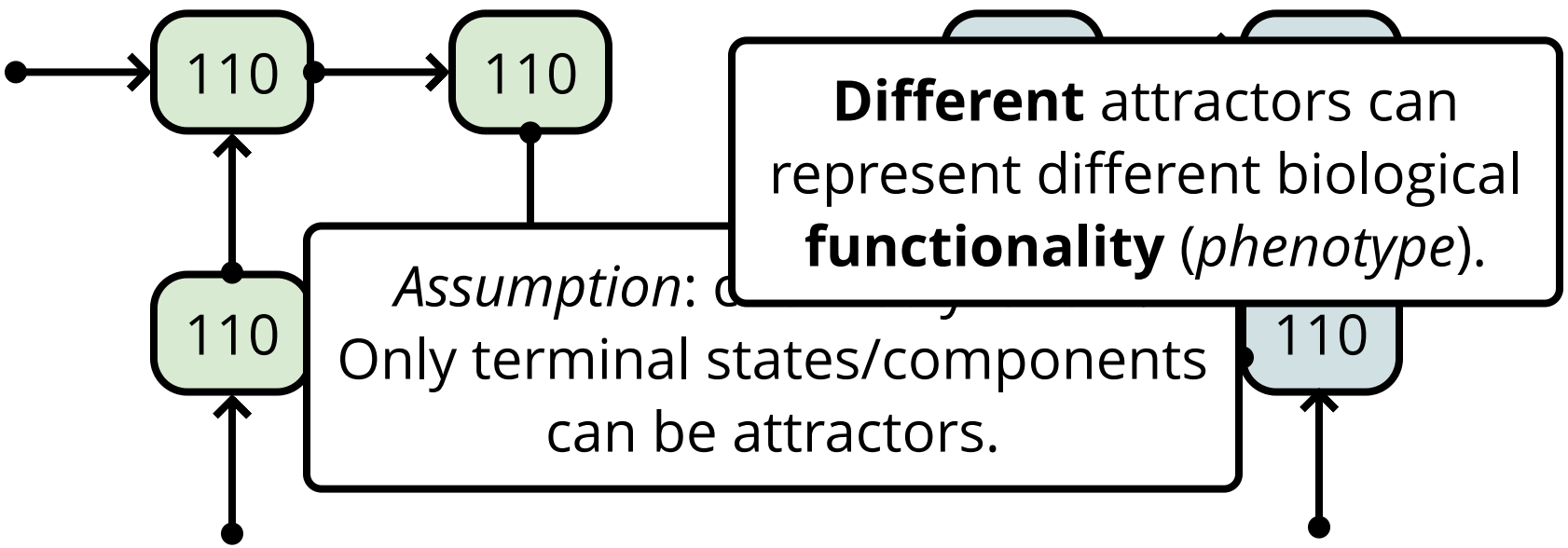




What "outcomes" are possible in our model? ***(Attractors - bottom SCCs)***

- Single state (*fixed-point attractor*)
- Larger region (*complex attractor*)
  - cycle (*oscillation*)
  - other (*disorder*)

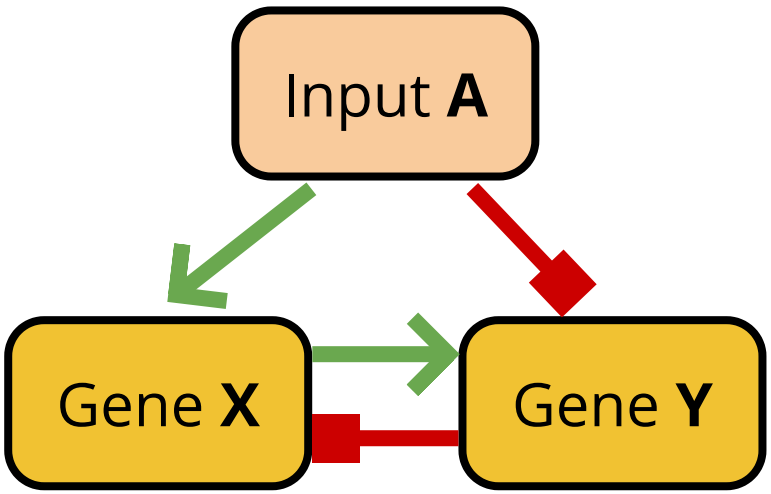




Addressing model uncertainty

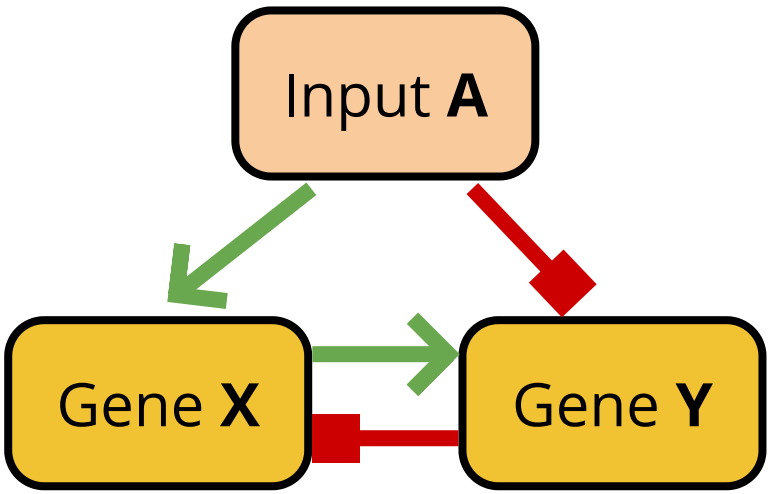
Addressing model uncertainty

- Logical parameters (network inputs)



Addressing model uncertainty

- Logical parameters (network inputs)



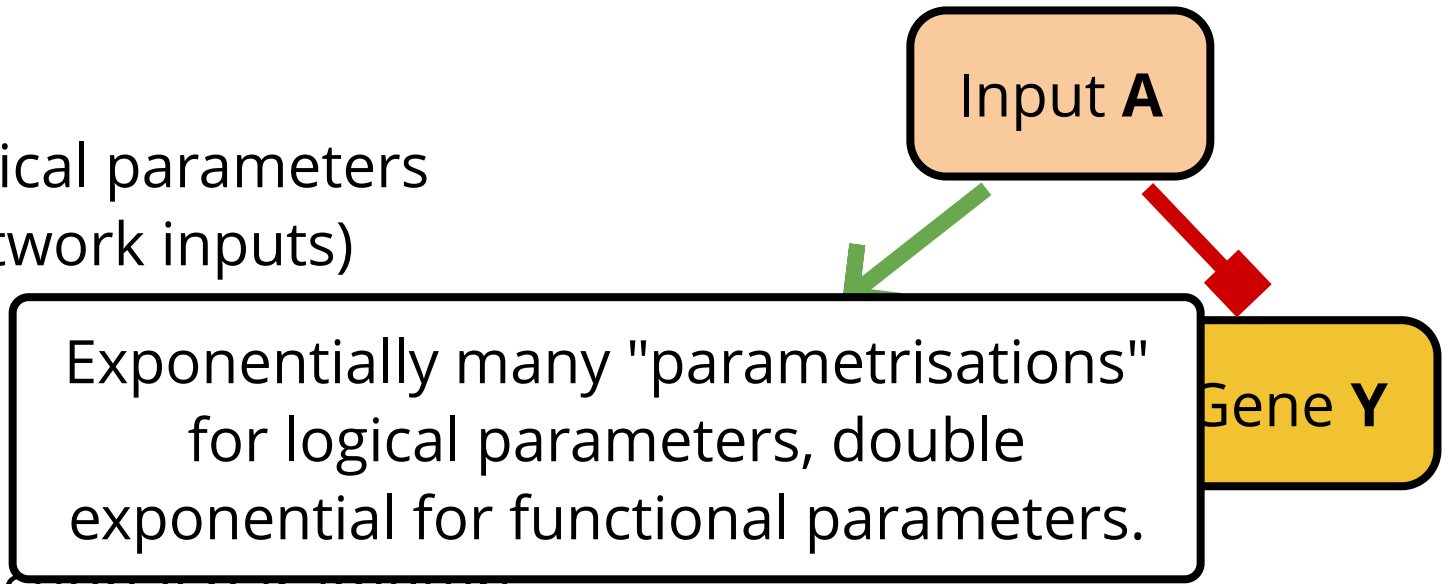
- Functional parameters (uninterpreted function)

$$f_C(x) = x_C \vee F(x_A, x_B)$$

Benes, N., Brim, L., Pastva, S., Polácek, J., & Šafránek, D. (2019). Formal Analysis of Qualitative Long-Term Behaviour in Parametrised Boolean Networks. *ICFEM*.

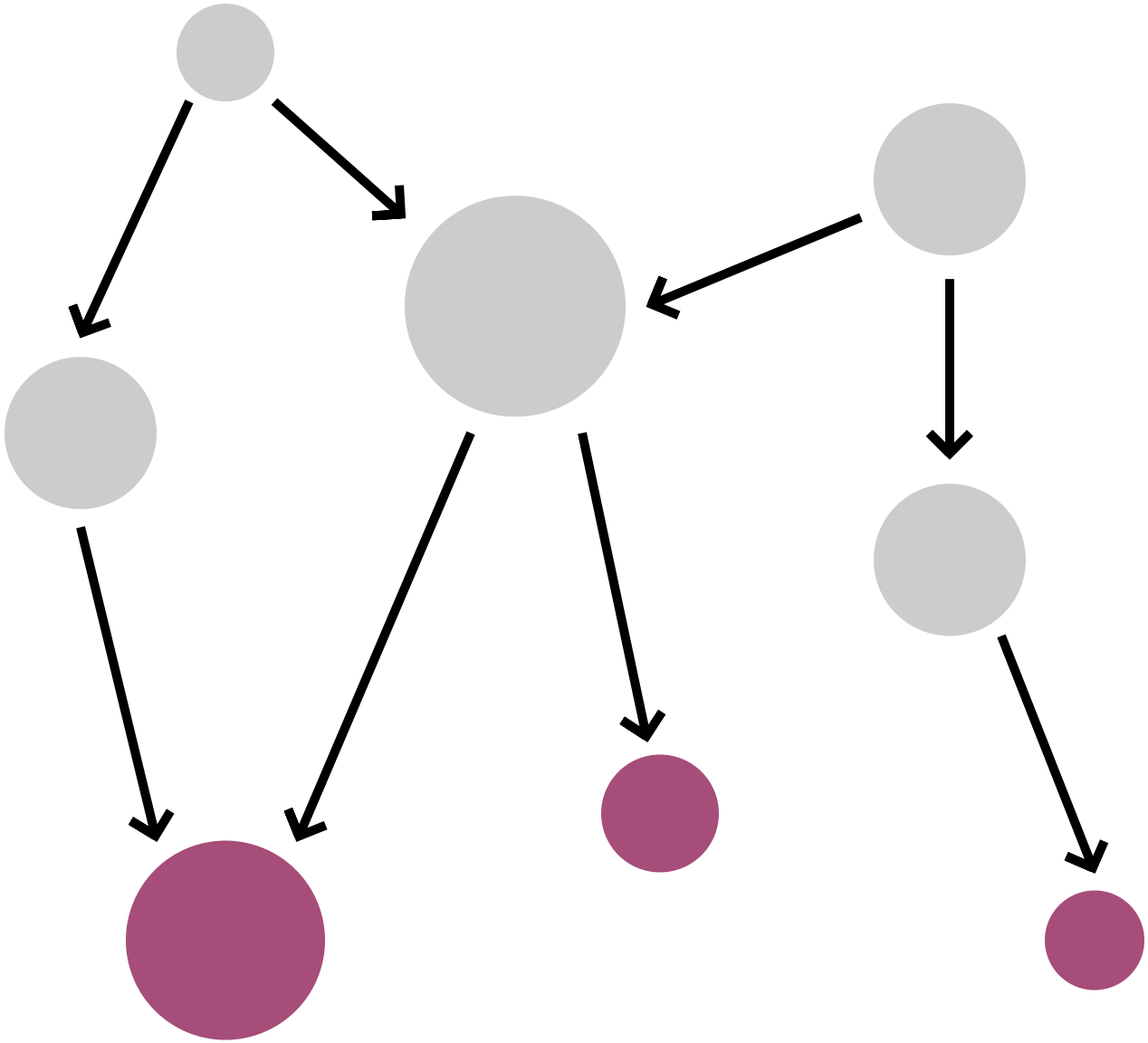
Addressing model uncertainty

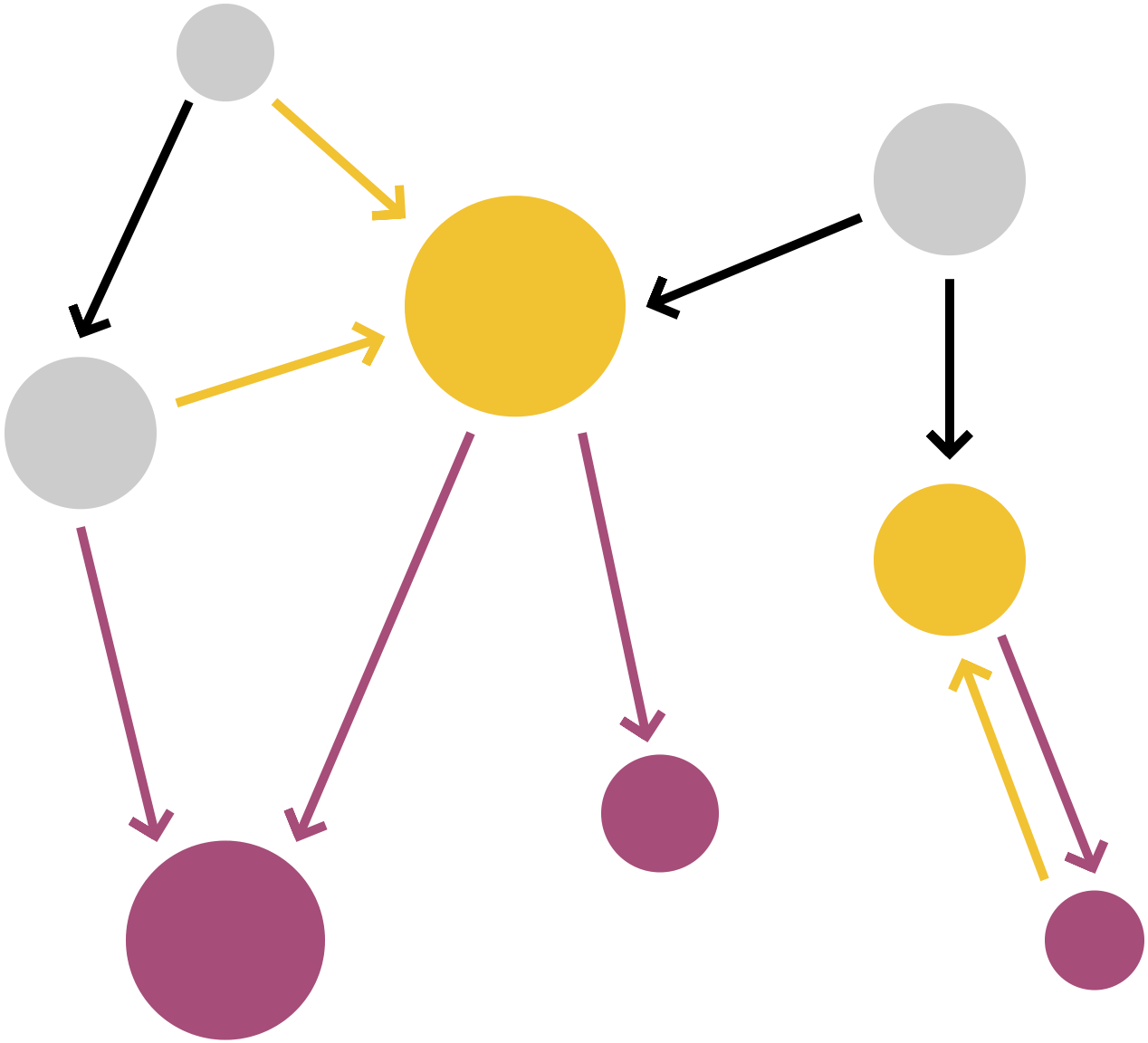
- Logical parameters (network inputs)
- Functional parameters (uninterpreted function)



$$f_C(x) = x_C \vee F(x_A, x_B)$$

Benes, N., Brim, L., Pastva, S., Polácek, J., & Šafránek, D. (2019). Formal Analysis of Qualitative Long-Term Behaviour in Parametrised Boolean Networks. *ICFEM*.





Control of Boolean networks

(The problems of)
Classical control theory for BNs

(The problems of)
Classical control theory for BNs

- *Non-determinism* is widespread;

(The problems of)
Classical control theory for BNs

- *Non-determinism* is widespread;
- Classical *stabilisation* may not be feasible (nor desired);

(The problems of)
Classical control theory for BNs

- *Non-determinism* is widespread;
- Classical *stabilisation* may not be feasible (nor desired);
- *Full* control is largely unnecessary;

(The problems of)
Classical control theory for BNs

- *Non-determinism* is widespread;
- Classical *stabilisation* may not be feasible (nor desired);
- *Full* control is largely unnecessary;
- *Output* control is more suitable, but can still include biologically unrealistic cases;

(The problems of)
Classical control theory for BNs

- *Non-determinism* is widespread;
- Classical *stabilisation* may not be feasible (nor desired);
- *Full* control is largely unnecessary;
- *Output* control is more suitable, but can still include biologically unrealistic cases;
- *Observability* is (in theory) high, but suffers from large delays;

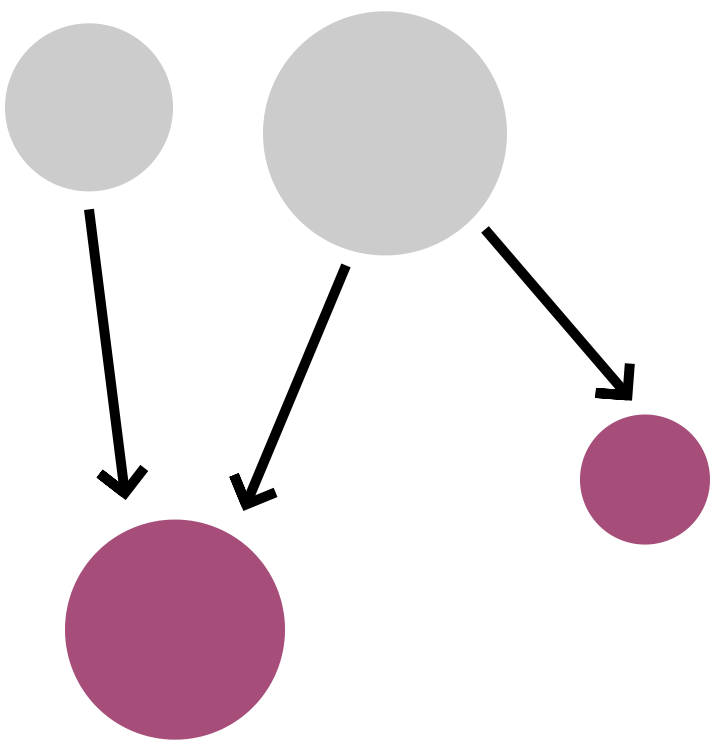
(The problems of)
Classical control theory for BNs

- *Non-determinism* is widespread;
- Classical *stabilisation* may not be feasible (nor desired);
- *Full* control is largely unnecessary;
- *Output* control is more suitable, but can still include biologically unrealistic cases;
- *Observability* is (in theory) high, but suffers from large delays;
- Every *intervention* is very costly;

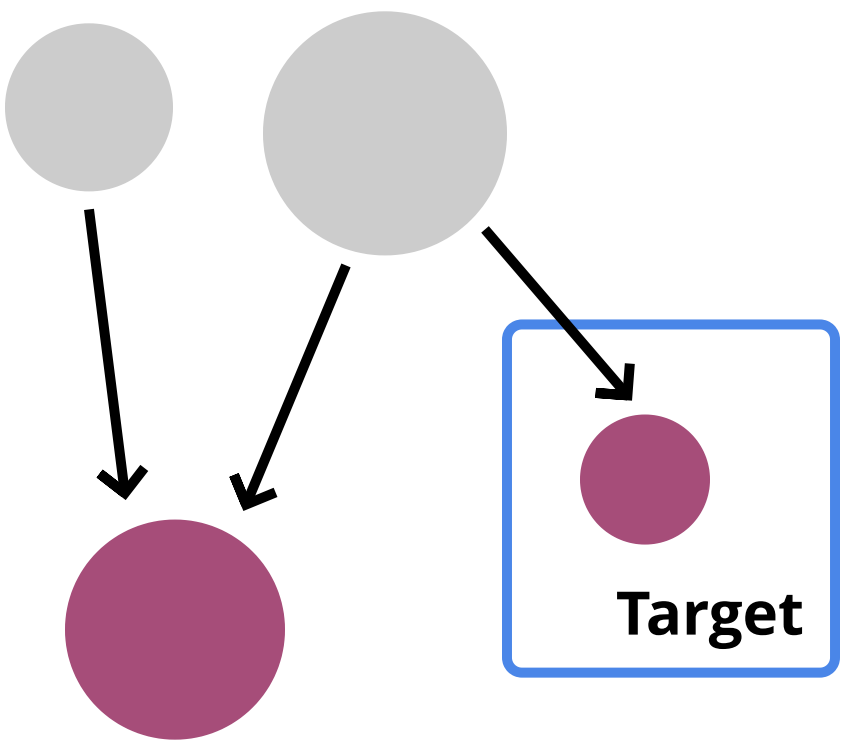
(The problems of)
Classical control theory for BNs

- *Non-determinism* is widespread;
- Classical *stabilisation* may not be feasible (nor desired);
- *Full* control is largely unnecessary;
- *Output* control is more suitable, but can still include biologically unrealistic cases;
- *Observability* is (in theory) high, but suffers from large delays;
- Every *intervention* is very costly;
- Interventions (mostly) cannot be applied indefinitely.

Boolean network "target" control



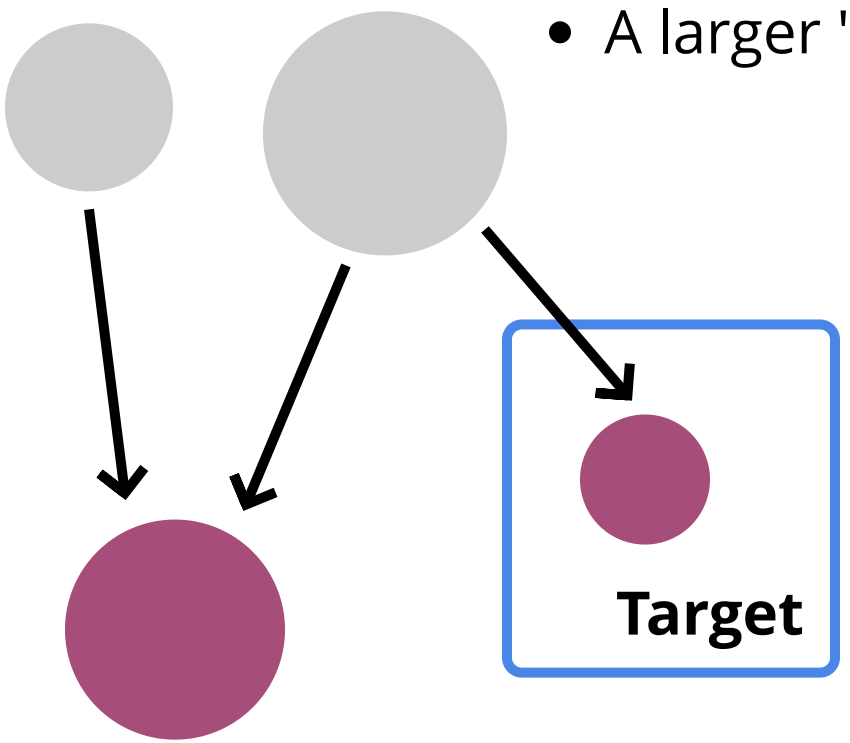
Boolean network "target" control



Boolean network "target" control

Target can mean slightly different things, depending on biological context, model, etc...

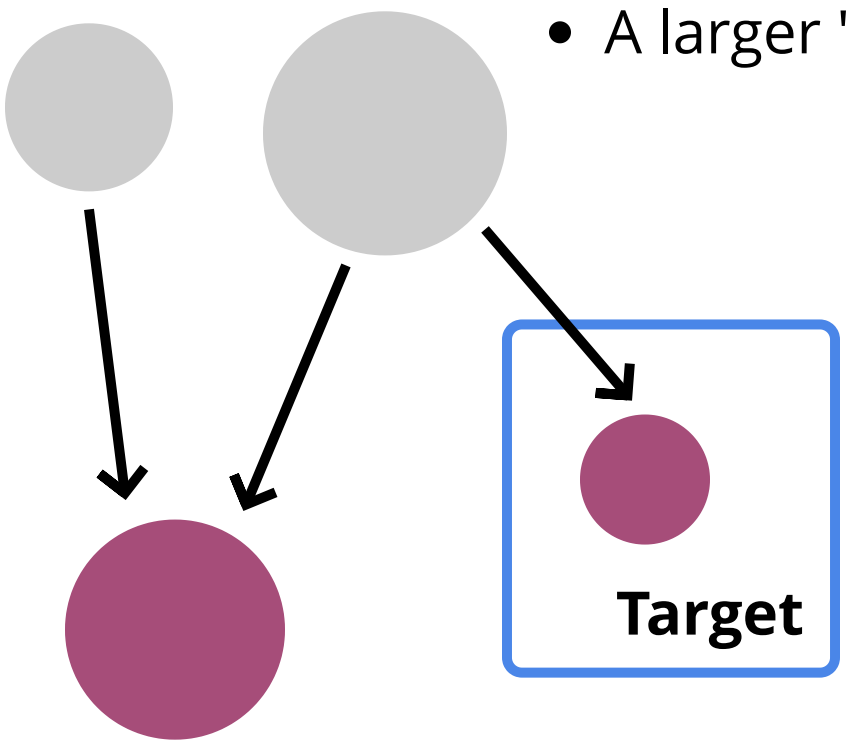
- A specific attractor(s) (exact set of states)
- Attractor state (must *appear* in the final attr.)
- A larger "subspace" (also called *phenotype*)



Boolean network "target" control

Target can mean slightly different things, depending on biological context, model, etc...

- A specific attractor(s) (exact set of states)
- Attractor state (must *appear* in the final attr.)
- A larger "subspace" (also called *phenotype*)

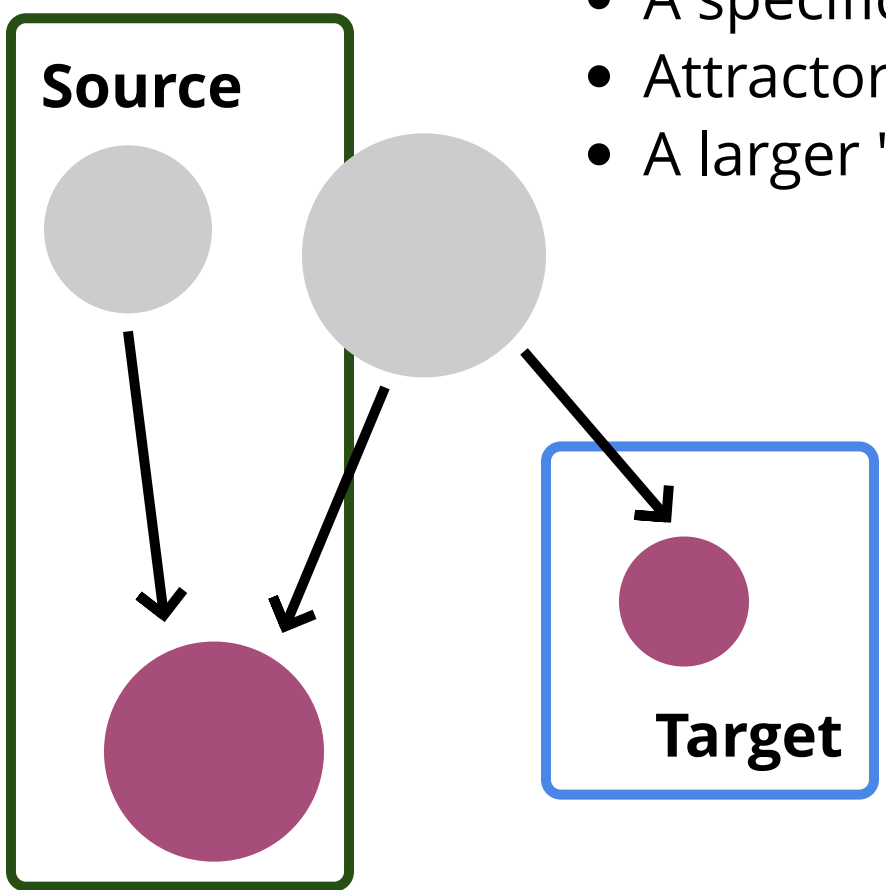


Regardless, *target* must admit some *trap set* in which the model can stay indefinitely.

Boolean network "target" control

Target can mean slightly different things, depending on biological context, model, etc...

- A specific attractor(s) (exact set of states)
- Attractor state (must *appear* in the final attr.)
- A larger "subspace" (also called *phenotype*)



Regardless, *target* must admit some *trap set* in which the model can stay indefinitely.

Optionally, *source* can be specified too: control must work in *all* source states.

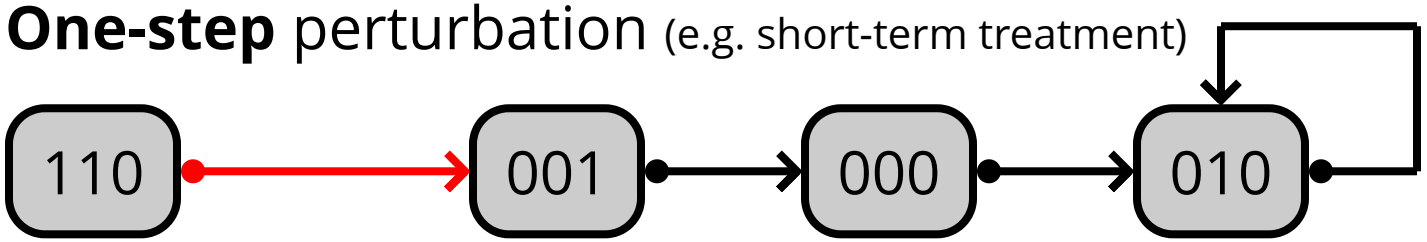
Controlling through perturbations

Perturbation forces a subset of variables
to a constant high/low state.

Controlling through perturbations

Perturbation forces a subset of variables
to a constant high/low state.

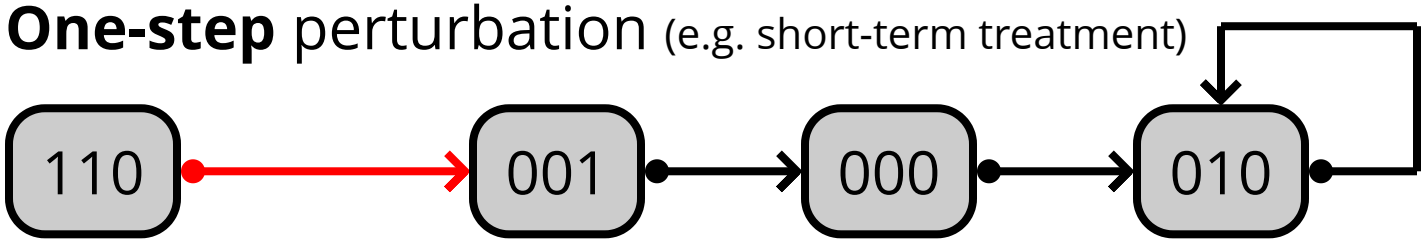
One-step perturbation (e.g. short-term treatment)



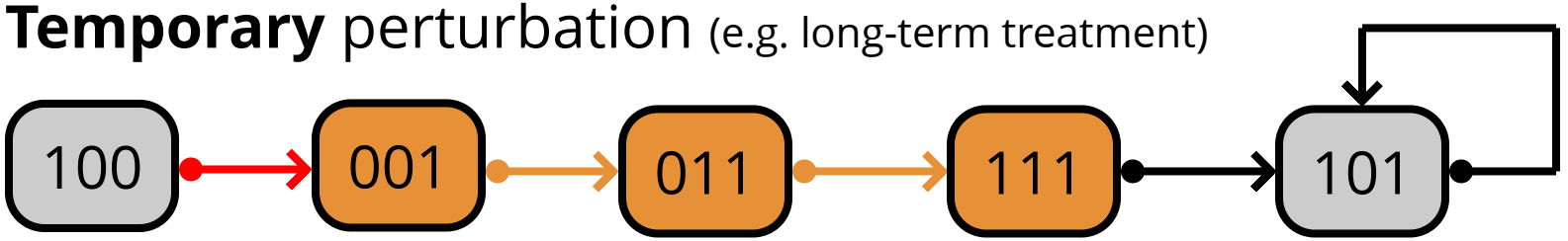
Controlling through perturbations

Perturbation forces a subset of variables
to a constant high/low state.

One-step perturbation (e.g. short-term treatment)



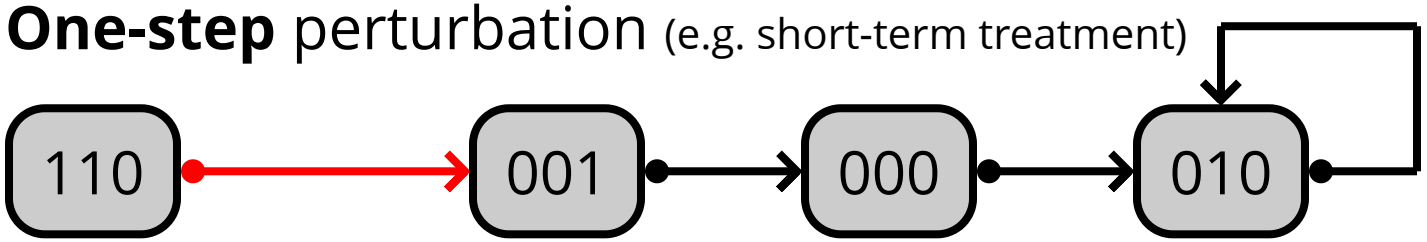
Temporary perturbation (e.g. long-term treatment)



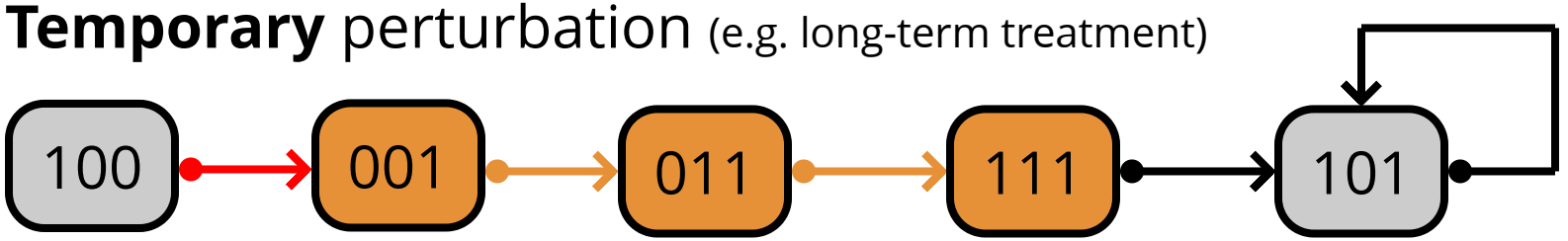
Controlling through perturbations

Perturbation forces a subset of variables
to a constant high/low state.

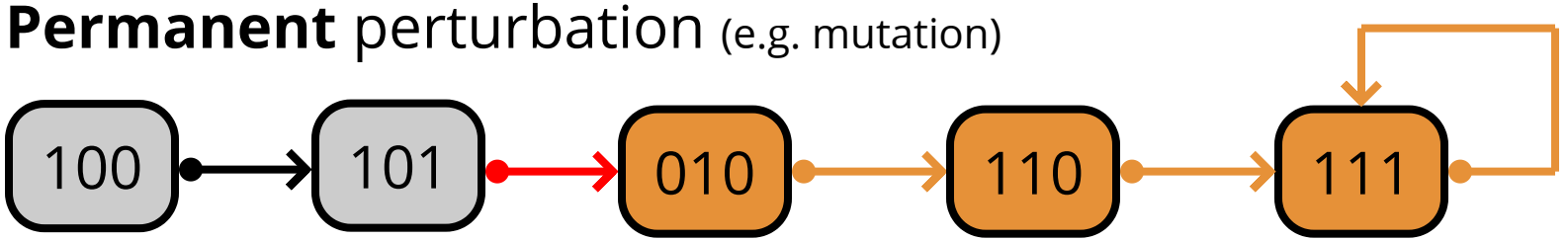
One-step perturbation (e.g. short-term treatment)

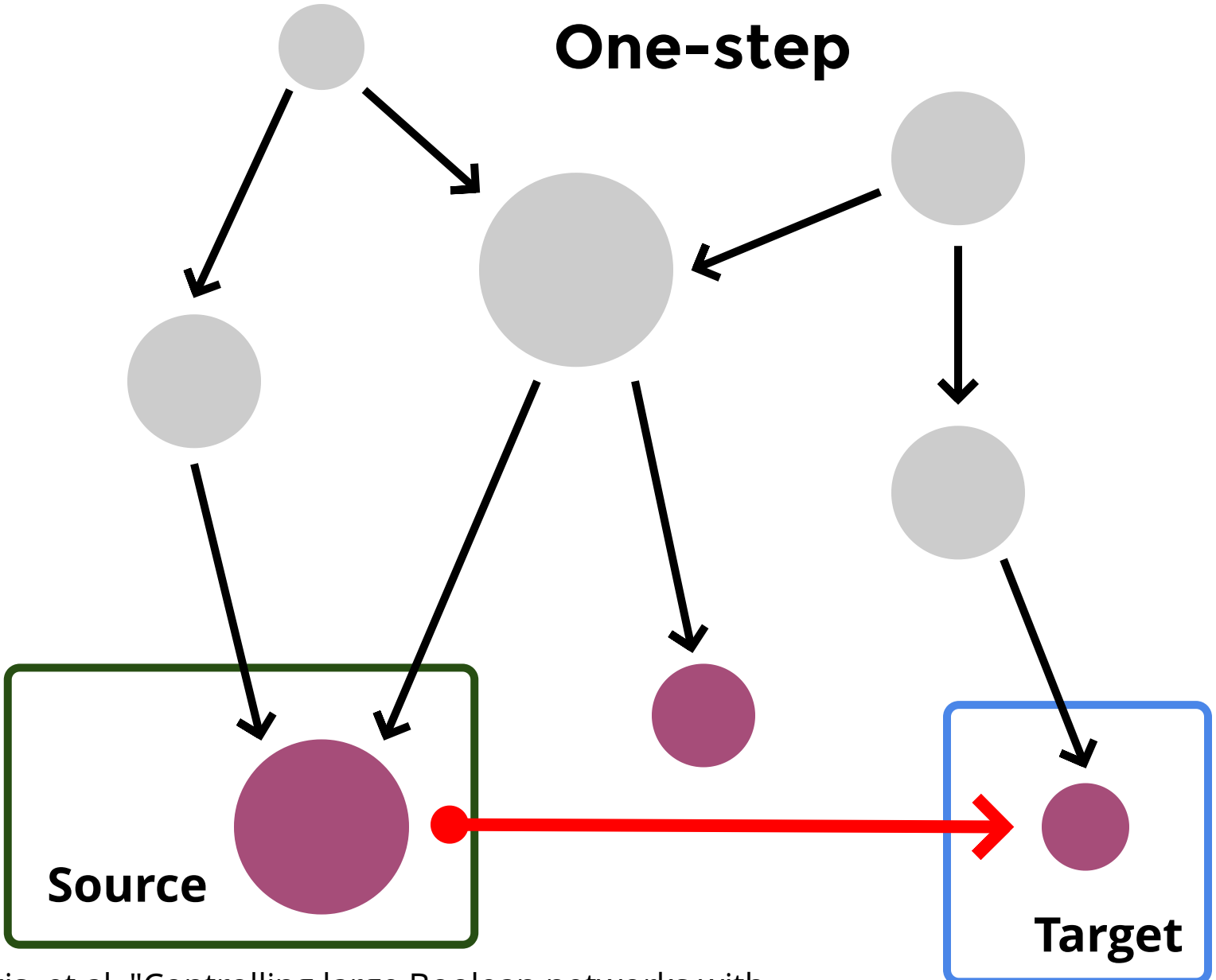


Temporary perturbation (e.g. long-term treatment)

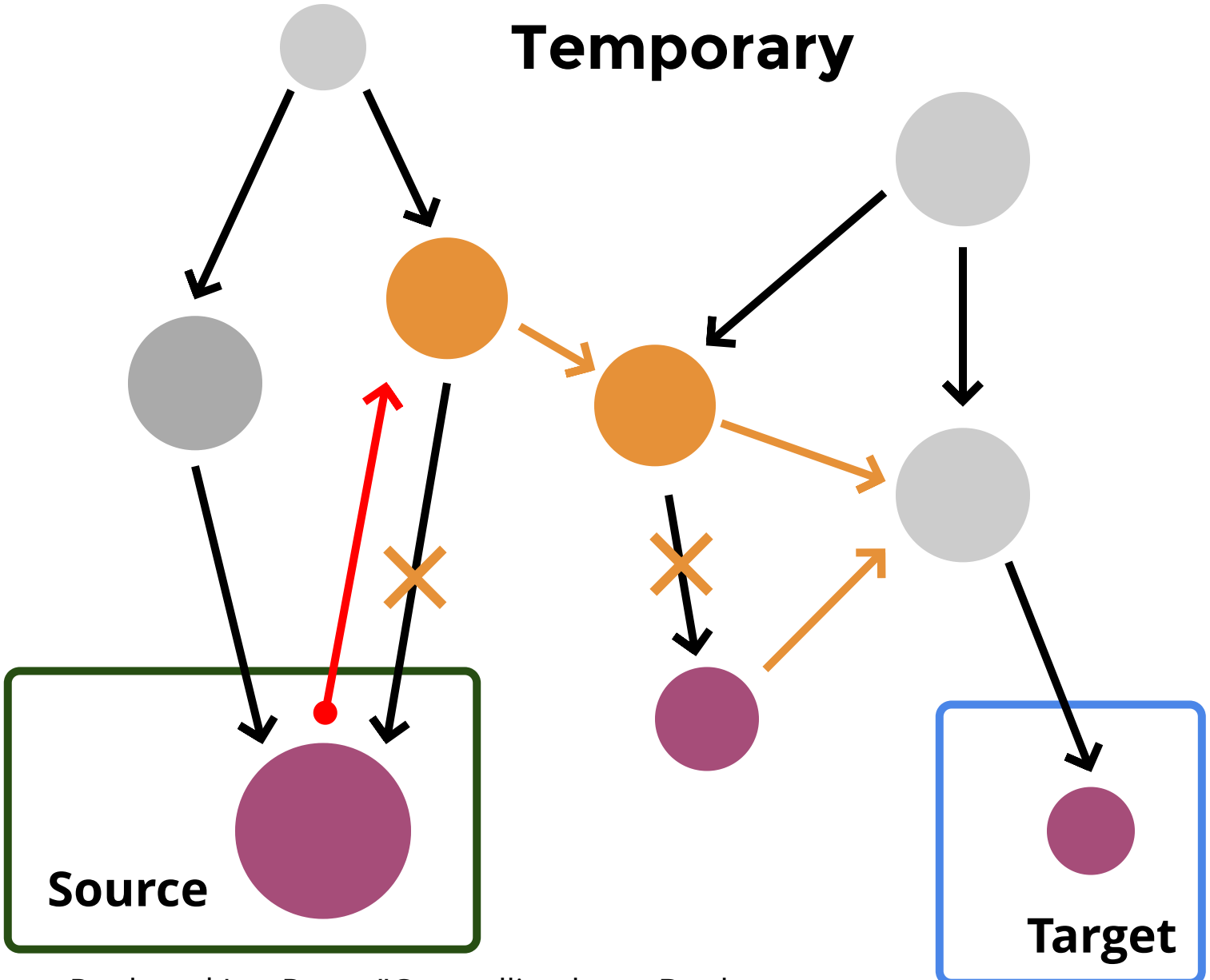


Permanent perturbation (e.g. mutation)

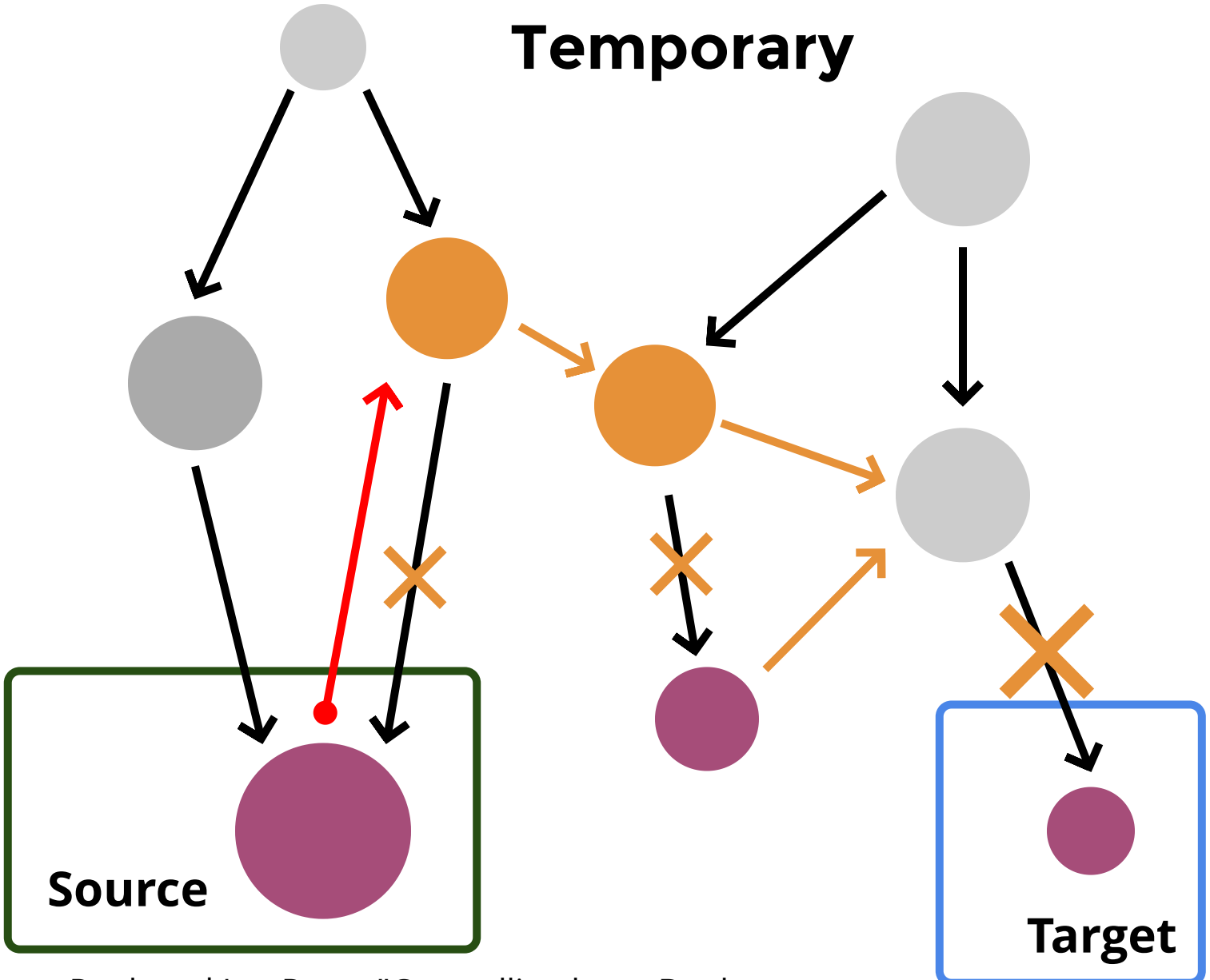




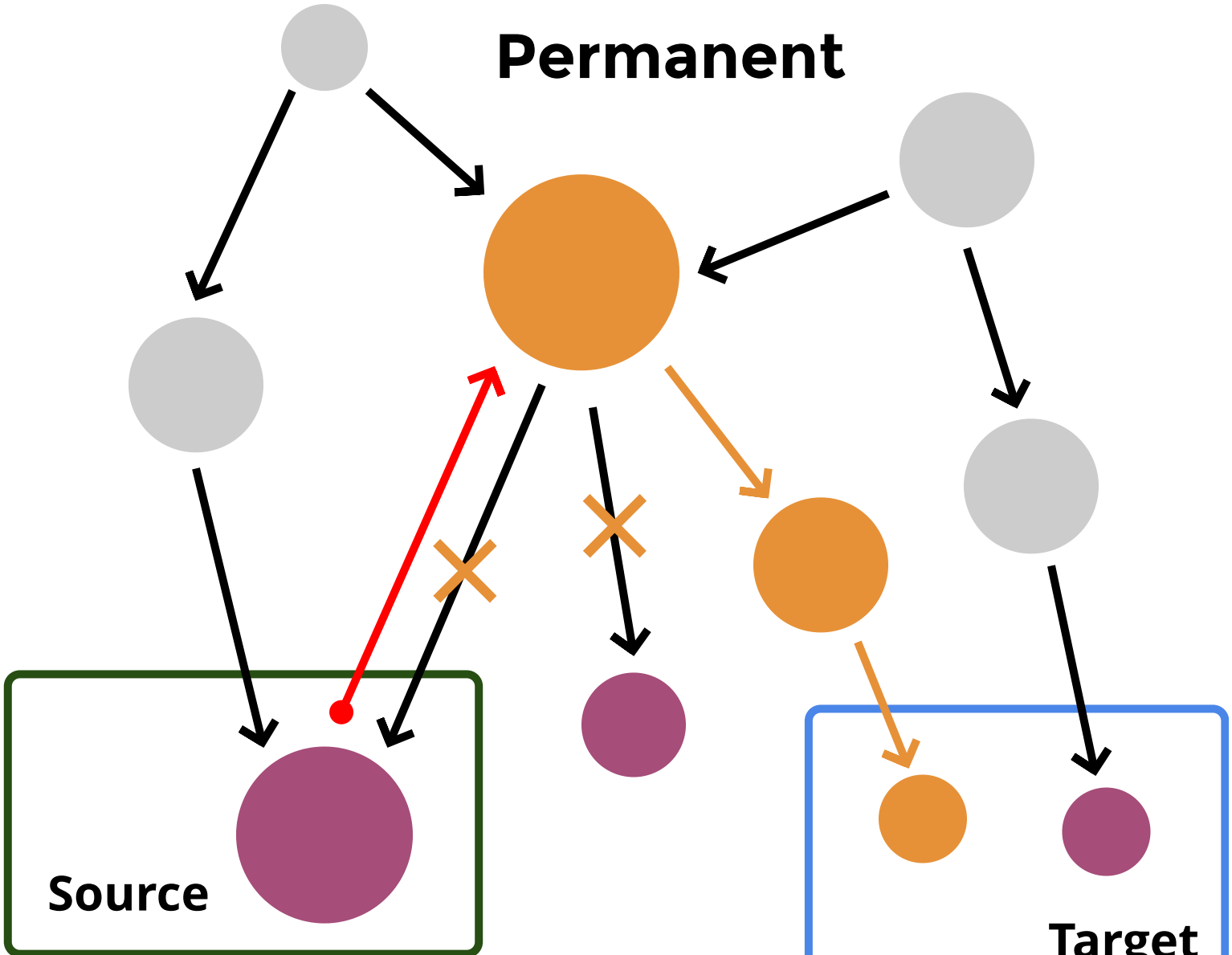
Baudin, Alexis, et al. "Controlling large Boolean networks with single-step perturbations." *Bioinformatics* 35.14 (2019): i558-i567.



Su, Cui, Soumya Paul, and Jun Pang. "Controlling large Boolean networks with temporary and permanent perturbations." *FM 2019.*



Su, Cui, Soumya Paul, and Jun Pang. "Controlling large Boolean networks with temporary and permanent perturbations." *FM 2019.*



Su, Cui, Soumya Paul, and Jun Pang. "Controlling large Boolean networks with temporary and permanent perturbations." *FM 2019.*

What perturbations do we actually want?

To minimize biological "burden" on the system, a perturbation must be as small (simple) as possible.

What perturbations do we actually want?

To minimize biological "burden" on the system, a perturbation must be as small (simple) as possible.

	Megakaryocyte	Erythrocyte	Granulocyte	Monocyte
Megakaryocyte	–	{EKLF} {Fli1}	{C/EBPα, PU1, Gfi1} {C/EBPα, Gfi1, GATA1}	{EgrNab, C/EBPα, GATA1} {EgrNab, C/EBPα, PU1}
Erythrocyte	{EKLF} {Fli1}	–	{C/EBPα, PU1, Gfi1} {C/EBPα, Gfi1, GATA1}	{EgrNab, C/EBPα, GATA1} {EgrNab, C/EBPα, PU1}
Granulocyte	{GATA2, Fli1} {GATA1, Fli1} {Fli1, PU1}	{GATA2, EKLF} {GATA1, EKLF}	–	{EgrNab} {Gfi1}
Monocyte	{GATA2, Fli1} {GATA1, Fli1} {Fli1, PU1}	{GATA2, EKLF} {GATA1, EKLF}	{PU1} {Gfi1},{cJun},{EgrNab}	–

In practice, *minimal* perturbations are often small!

What perturbations do we actually want?

To minimize biological "burden" on the system, a perturbation must be as small (simple) as possible.

	Megakaryocyte	Erythrocyte	Granulocyte	Monocyte
Megakaryocyte	–	{EKLF} {Fli1}	{C/EBPα, PU1, Gfi1} {C/EBPα, Gfi1, GATA1}	{EgrNab, C/EBPα, GATA1} {EgrNab, C/EBPα, PU1}
Erythrocyte	{EKLF} {Fli1}	–	{C/EBPα, PU1, Gfi1} {C/EBPα, Gfi1, GATA1}	{EgrNab, C/EBPα, GATA1} {EgrNab, C/EBPα, PU1}
Granulocyte	{GATA2, Fli1} {GATA1, Fli1} {Fli1, PU1}	{GATA2, EKLF} {GATA1, EKLF}	–	{EgrNab} {Gfi1}
Monocyte	{GATA2, Fli1} {GATA1, Fli1} {Fli1, PU1}	{GATA2, EKLF} {GATA1, EKLF}	{PU1} {Gfi1},{cJun},{EgrNab}	–

In practice, *minimal* perturbations are often small!

Additional steps are needed to asses biological viability of perturbations!

What about partially unknown networks?

What about partially unknown networks?

Whether a perturbation works depends on the valuation of logical and functional parameters.

What about partially unknown networks?

Whether a perturbation works depends on the valuation of logical and functional parameters.

A small perturbation that works only sometimes may be too unreliable.

What about partially unknown networks?

Whether a perturbation works depends on the valuation of logical and functional parameters.

A small perturbation that works only sometimes

may be too unreliable.

A large perturbation that works always

may not be biologically viable.

Perturbation robustness

$$\text{robustness}(p) = \frac{\text{\# parameter valuations where perturbation } p \text{ achieves control}}{\text{\# parameter valuations that admit the desired target}}$$

Brim, Luboš, et al. "Temporary and permanent control of partially specified Boolean networks." *Biosystems* 223 (2023): 104795.

Perturbation robustness

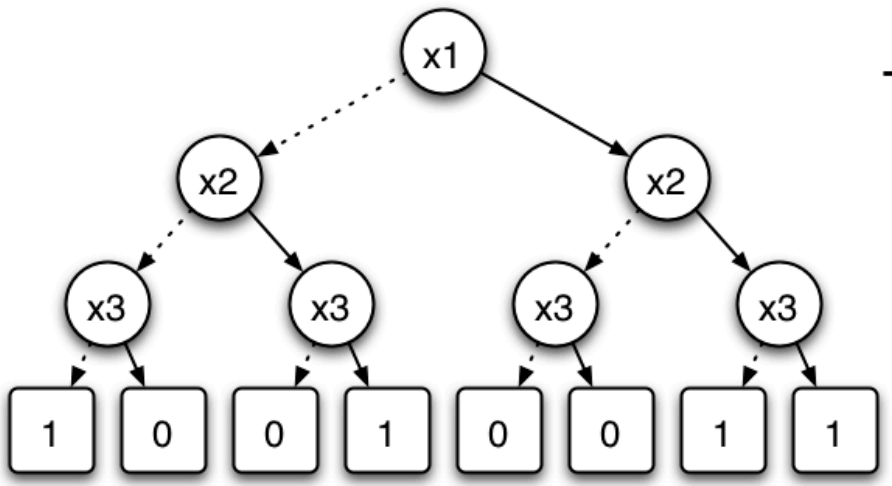
$$\text{robustness}(p) = \frac{\#\text{ parameter valuations where perturbation } p \text{ achieves control}}{\#\text{ parameter valuations that admit the desired target}}$$

A good perturbation is not only *small*, but also *robust*.

Brim, Luboš, et al. "Temporary and permanent control of partially specified Boolean networks." *Biosystems* 223 (2023): 104795.

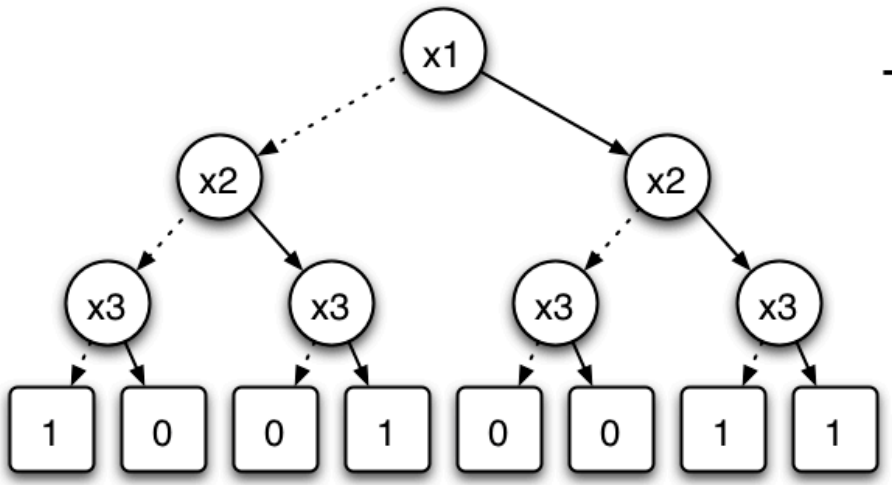
How to compute robust perturbations?

Binary Decision Diagrams

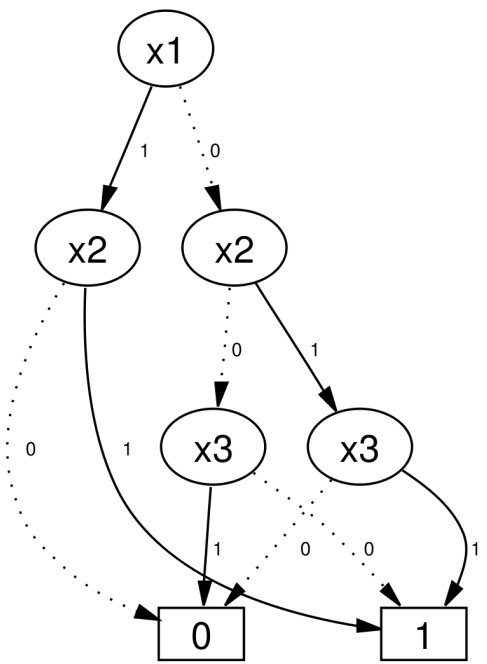


x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

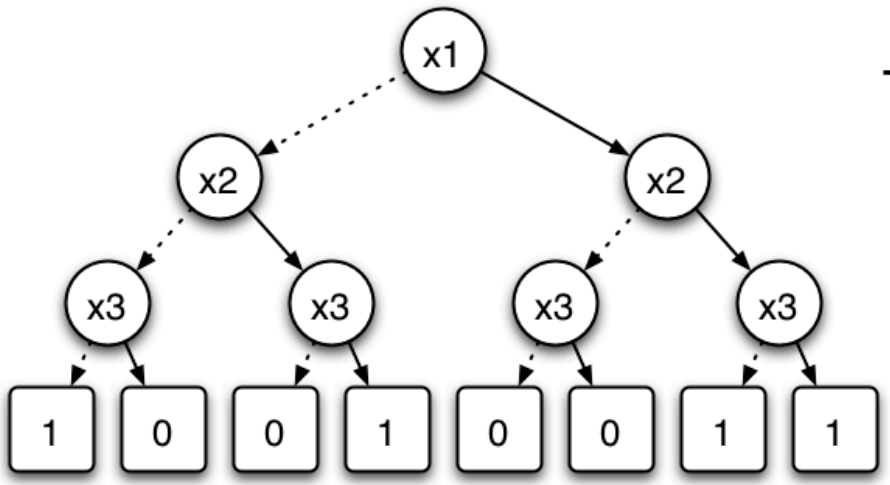
Binary Decision Diagrams



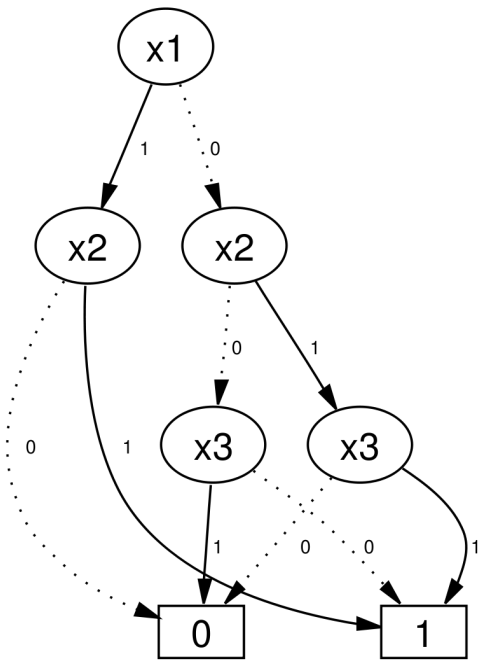
x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



Binary Decision Diagrams



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



Parametrisation+State of a BN is a bit-vector. Sets/relations of these can be encoded via Boolean functions.

$$s \in X \Leftrightarrow f_{BDD(X)}(s) = 1 \,(\text{with } X \subseteq \{0,1\}^k)$$

$$X \cap Y \equiv f_{BDD(X)} \wedge f_{BDD(Y)}$$

Set operations correspond to logical operations on functions (and graph operations on BDDs)

$$X \cap Y \equiv f_{BDD(X)} \wedge f_{BDD(Y)}$$

Set operations correspond to logical operations on functions (and graph operations on BDDs)

$$X \cap Y \equiv f_{BDD(X)} \wedge f_{BDD(Y)}$$

Since Boolean network is described by Boolean functions, computing successors/predecessors *(including parameters)* is relatively straightforward.

Benes, N., Brim, L., Pastva, S., Polácek, J., & Šafránek, D. (2019). Formal Analysis of Qualitative Long-Term Behaviour in Parametrised Boolean Networks. *ICFEM*.

Symbolic encoding of perturbations

A perturbation is treated as another type of parameter:

Brim, Luboš, et al. "Temporary and permanent control of partially specified Boolean networks." *Biosystems* 223 (2023): 104795.

Symbolic encoding of perturbations

A perturbation is treated as another type of parameter:

$$f_C(x) = x_C \vee F(x_A, x_B)$$

Brim, Luboš, et al. "Temporary and permanent control of partially specified Boolean networks." *Biosystems* 223 (2023): 104795.

Symbolic encoding of perturbations

A perturbation is treated as another type of parameter:

$$f_C(x) = x_C \vee F(x_A, x_B)$$

$$\hat{f}_C(x) = (p_C \Rightarrow x_C) \wedge (\neg p_C \Rightarrow f_C(x))$$

Brim, Luboš, et al. "Temporary and permanent control of partially specified Boolean networks." *Biosystems* 223 (2023): 104795.

Symbolic encoding of perturbations

A perturbation is treated as another type of parameter:

$$f_C(x) = x_C \vee F(x_A, x_B)$$

$$\hat{f}_C(x) = (p_C \Rightarrow x_C) \wedge (\neg p_C \Rightarrow f_C(x))$$

"Perturbation parameters" need to be differentiated from "uncertain" parameters (i.e. during robustness assessment).

Brim, Luboš, et al. "Temporary and permanent control of partially specified Boolean networks." *Biosystems* 223 (2023): 104795.

Symbolic encoding of perturbations

A perturbation is treated as another type of parameter:

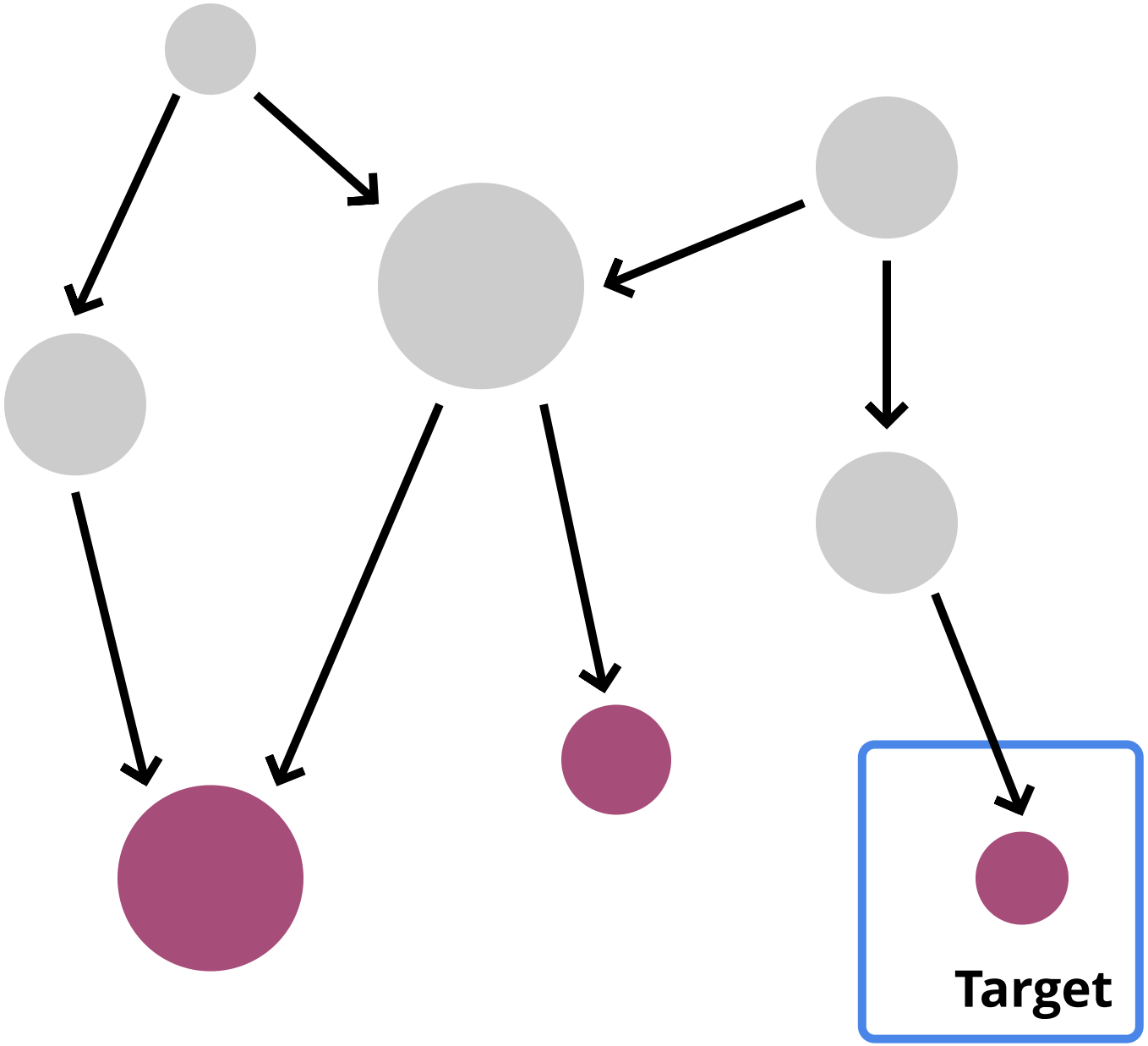
$$f_C(x) = x_C \vee F(x_A, x_B)$$

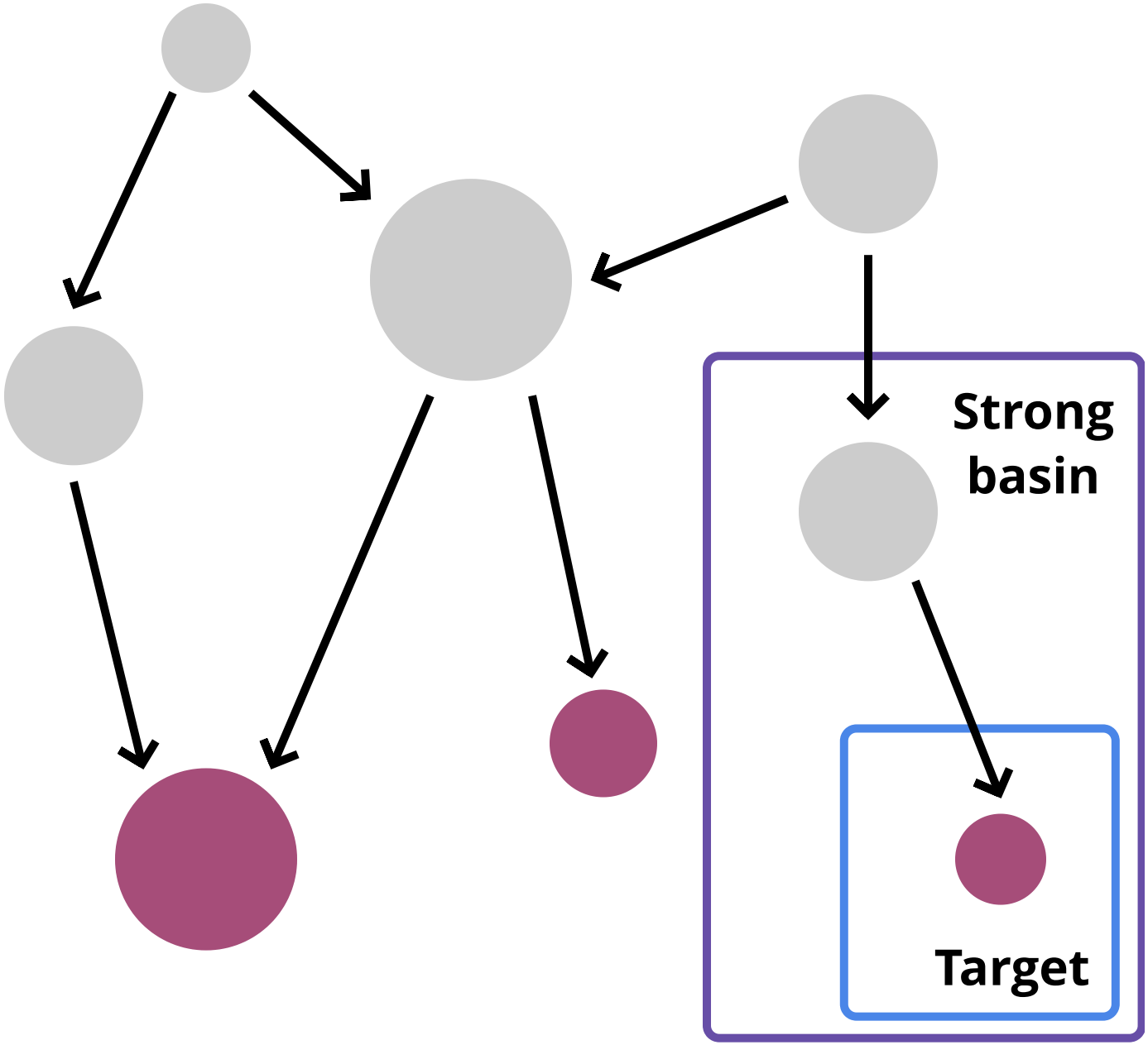
$$\hat{f}_C(x) = (p_C \Rightarrow x_C) \wedge (\neg p_C \Rightarrow f_C(x))$$

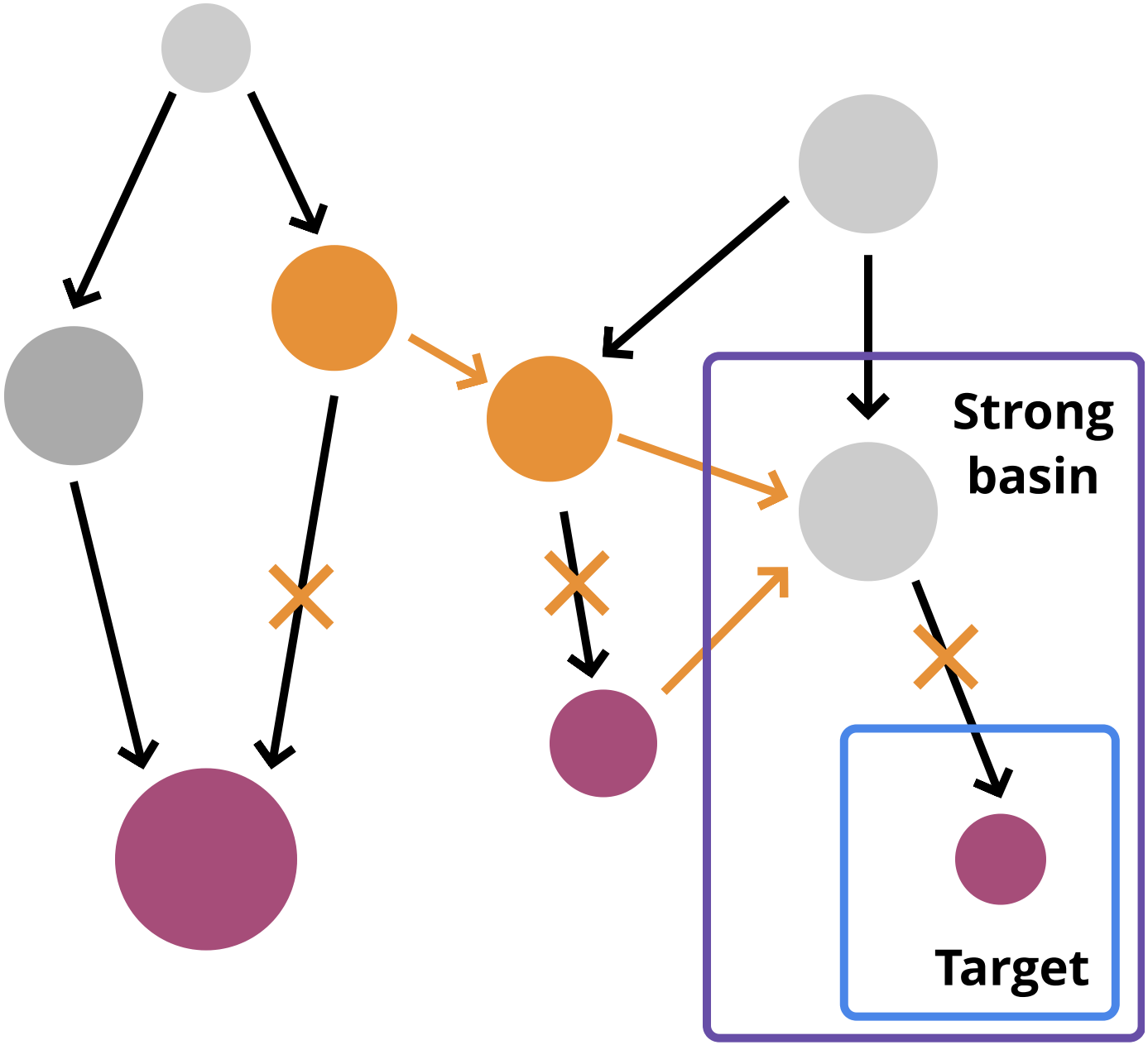
"Perturbation parameters" need to be differentiated from "uncertain" parameters (i.e. during robustness assessment).

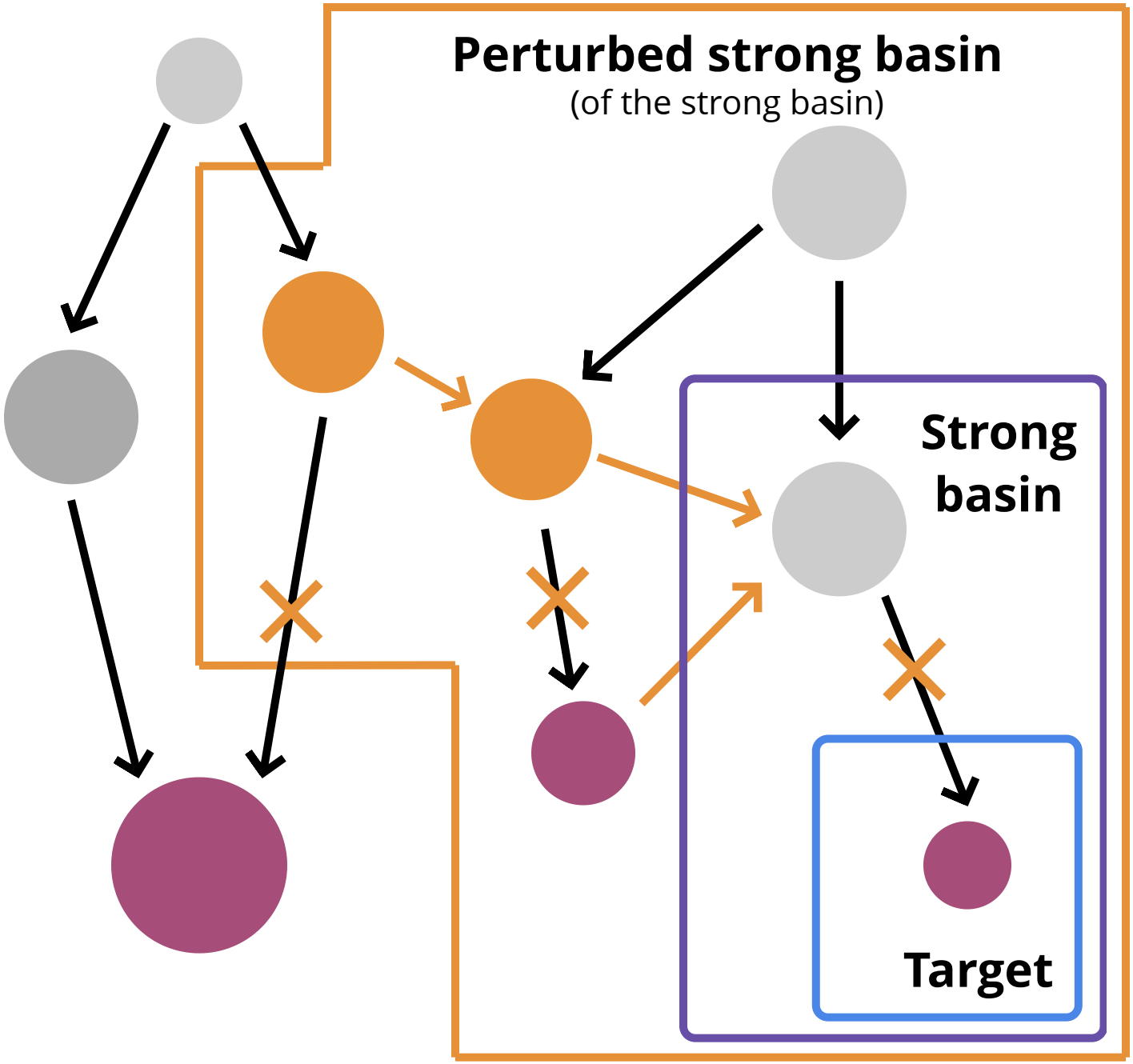
"Naive" encoding uses two "bits" per variable. Here we show that algorithms also exist for a simpler "one bit" encoding.

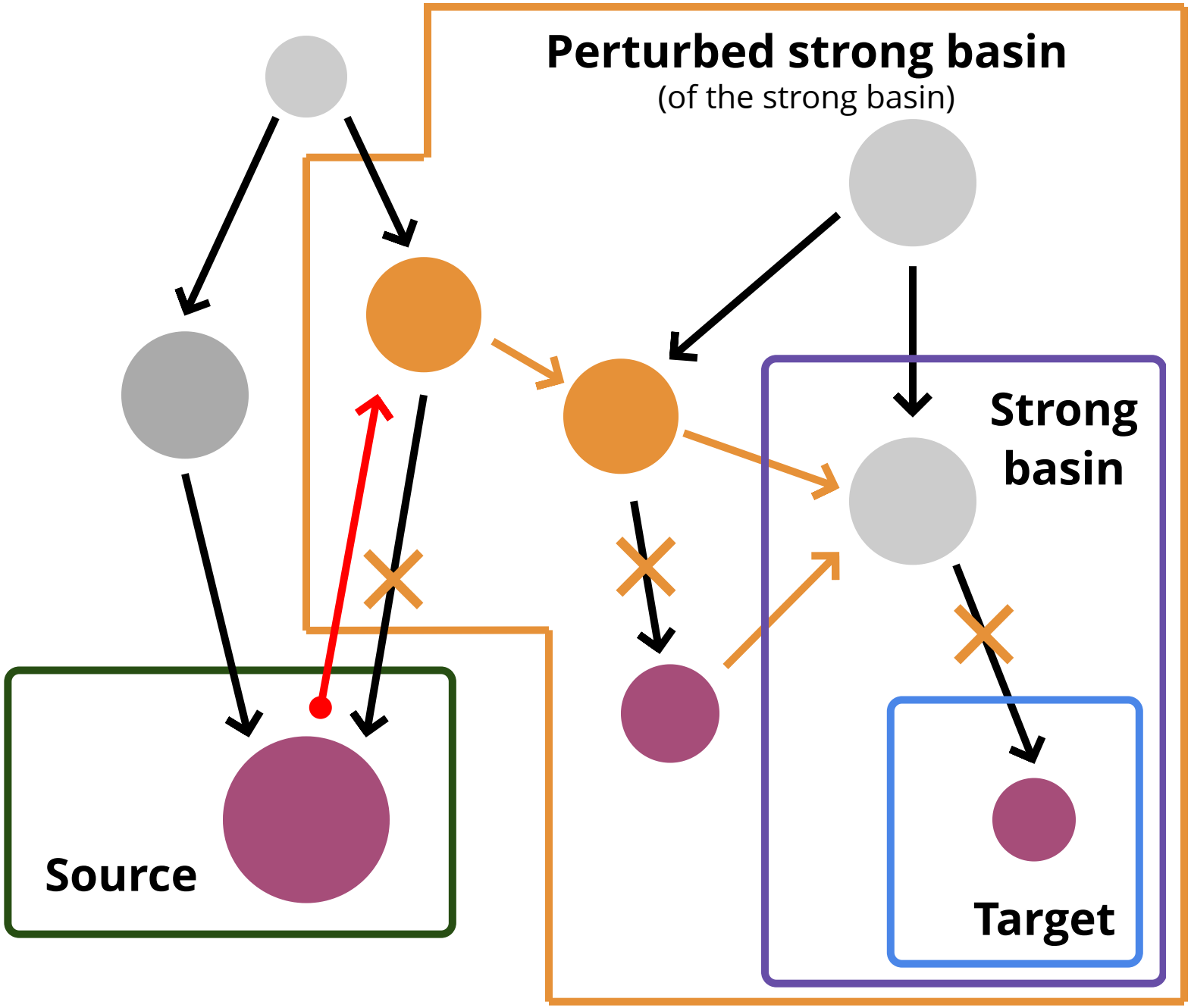
Brim, Luboš, et al. "Temporary and permanent control of partially specified Boolean networks." *Biosystems* 223 (2023): 104795.


Target


Strong
basin
Target


Strong
basin
Target


Perturbed strong basin
(of the strong basin)
Strong basin
Target


Perturbed strong basin
(of the strong basin)
Strong basin
Source
Target

Back to robustness...

Algorithm runs symbolically for all perturbations and parametrisations.

Result: A symbolic relation (a BDD) with all viable perturbations for all viable parametrisations.

Back to robustness...

Algorithm runs symbolically for all perturbations and parametrisations.

Result: A symbolic relation (a BDD) with all viable perturbations for all viable parametrisations.

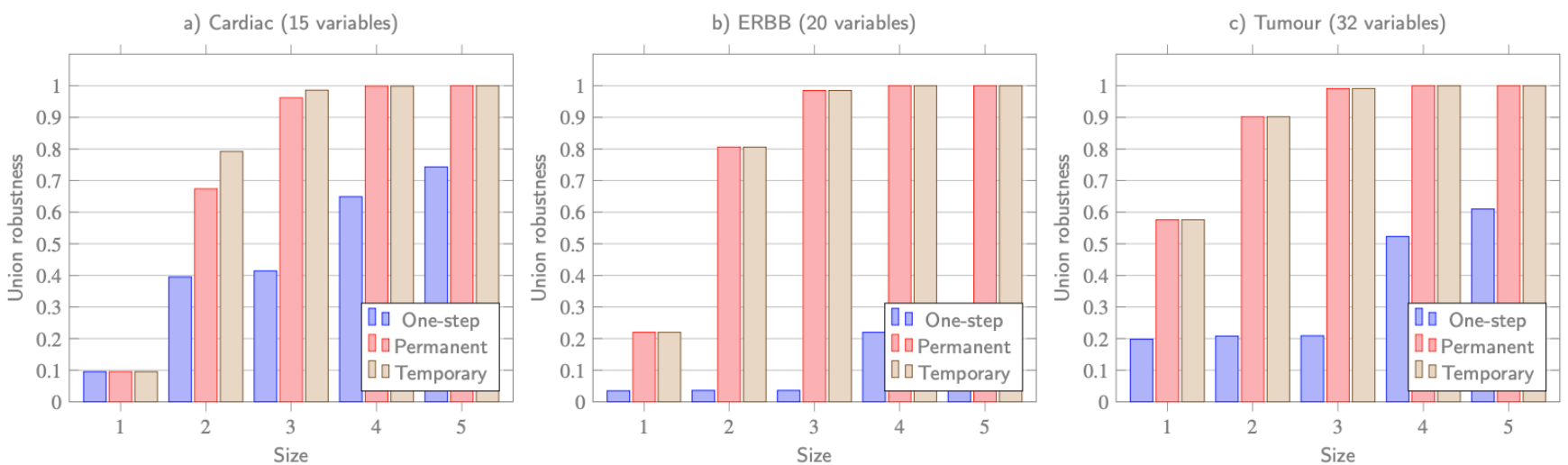
We can "search" this relation to find minimal perturbations, robust perturbations, or both.

Back to robustness...

Algorithm runs symbolically for all perturbations and parametrisations.

Result: A symbolic relation (a BDD) with all viable perturbations for all viable parametrisations.

We can "search" this relation to find minimal perturbations, robust perturbations, or both.



*up to 10^9 parameter valuations/model

Conclusion

Conclusion

- *Classical control theory* is not well equipped to handle "logical" models in systems biology.

Conclusion

- *Classical control theory* is not well equipped to handle "logical" models in systems biology.
- Various notions of *target* (or source-target) control are evolving to cover this need.

Conclusion

- *Classical control theory* is not well equipped to handle "logical" models in systems biology.
- Various notions of *target* (or source-target) control are evolving to cover this need.
- Here, *small* perturbations are typically sufficient to achieve the desired target.

Conclusion

- *Classical control theory* is not well equipped to handle "logical" models in systems biology.
- Various notions of *target* (or source-target) control are evolving to cover this need.
- Here, *small* perturbations are typically sufficient to achieve the desired target.
- However, *uncertainty* is often needed in biological models ⇒ partially specified Boolean networks.

Conclusion

- *Classical control theory* is not well equipped to handle "logical" models in systems biology.
- Various notions of *target* (or source-target) control are evolving to cover this need.
- Here, *small* perturbations are typically sufficient to achieve the desired target.
- However, *uncertainty* is often needed in biological models ⇒ partially specified Boolean networks.
- A good perturbation is not only small, but also *robust*.

Conclusion

- *Classical control theory* is not well equipped to handle "logical" models in systems biology.
- Various notions of *target* (or source-target) control are evolving to cover this need.
- Here, *small* perturbations are typically sufficient to achieve the desired target.
- However, *uncertainty* is often needed in biological models ⇒ partially specified Boolean networks.
- A good perturbation is not only small, but also *robust*.
- *Symbolic methods* are a viable starting point, but further research is needed!

Conclusion

- *Classical control theory* is not well equipped to handle "logical" models in systems biology.
- Various notions of *target* (or source-target) control are evolving to cover this need.
- Here, *small* perturbations are typically sufficient to achieve the desired target.
- However, *uncertainty* is often needed in biological models ⇒ partially specified Boolean networks.
- A good perturbation is not only small, but also *robust*.
- *Symbolic methods* are a viable starting point, but further research is needed!
- Minimal perturbations typically aren't robust, but robust perturbations can still be small.