Abstract Interpretation for Traces of Automata Networks

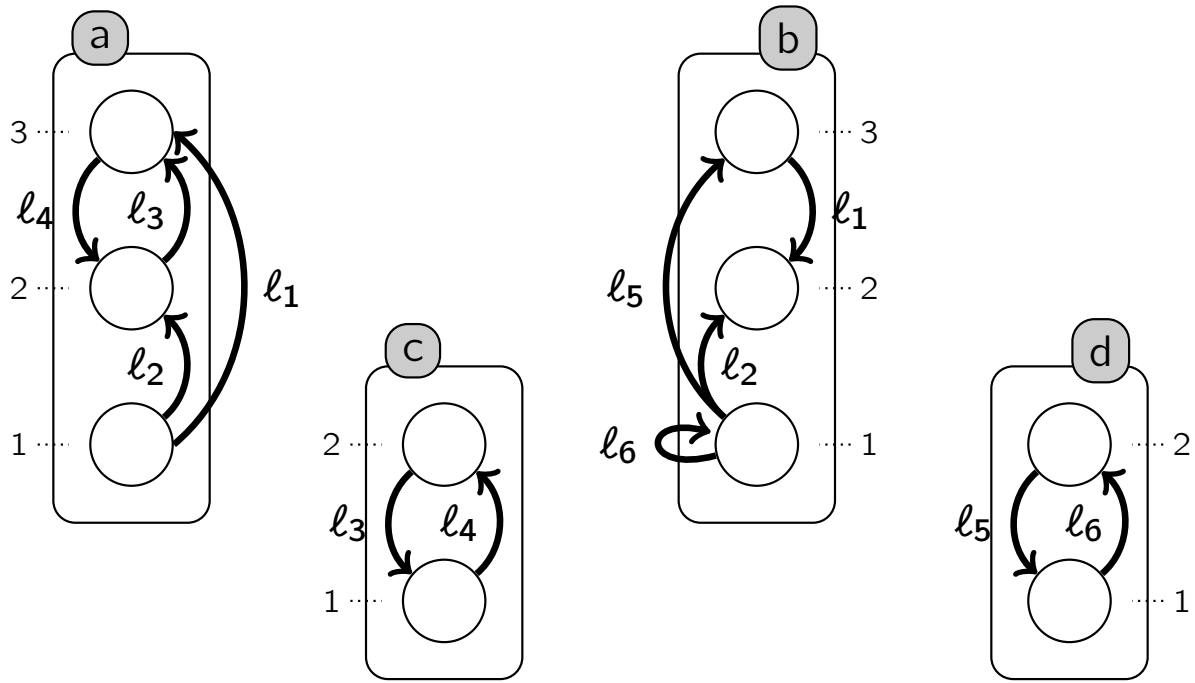
Loïc Paulevé

CNRS / LRI, Université Paris-Sud, France (Bioinfo team)
loic.pauleve@lri.fr
http://loicpauleve.name

Joint work with *Olivier Roux*, *Morgan Magnin*, *Maxime Folschette* (IRCCyN, Nantes), *Geoffroy Andrieux* (IRISA, Rennes), *Heinz Koeppl* (ETH Zürich)

Journée AFSEC Concurrence
November 19, 2013, ENS Cachan, France

Finite Automata Networks

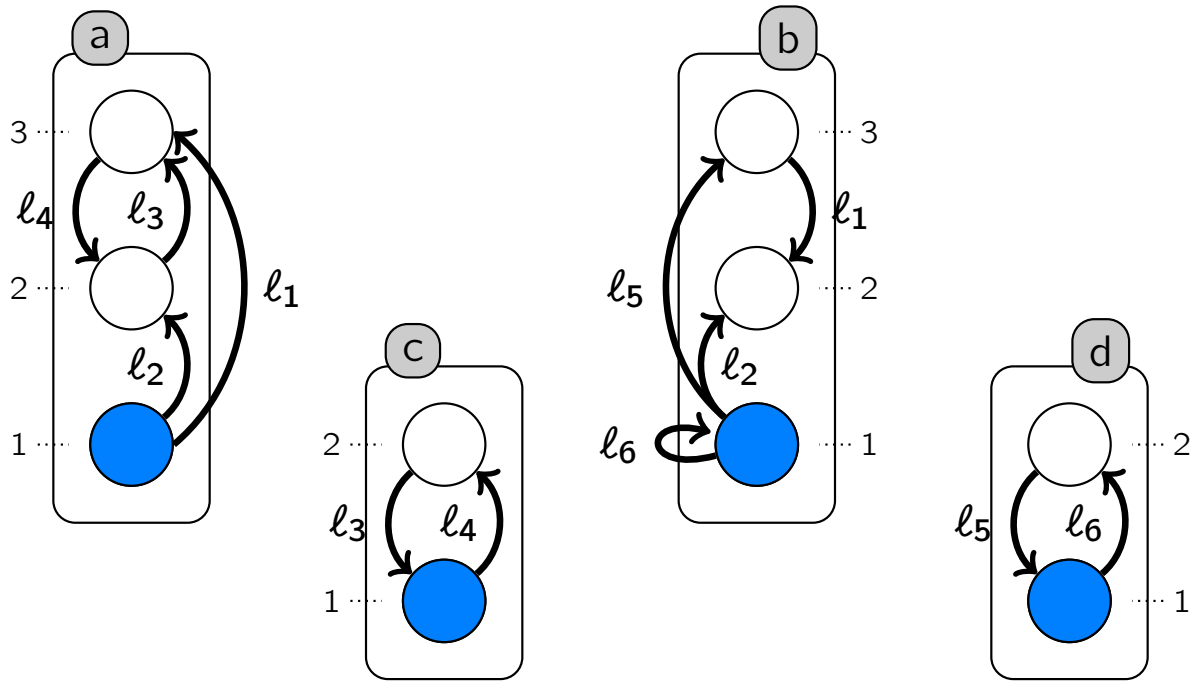


- transition ℓ pre-condition: ${}^\bullet\ell = \{a_i \mid a_i \xrightarrow{\ell} a_j\}$:

$$s \to s' \overset{\Delta}{\Leftrightarrow} \exists \ell \in \mathcal{L} : \forall a_i \in {}^\bullet\ell, s(a) = a_i \wedge \forall a_j \in \ell^\bullet, s'(a) = a_j$$
$$\wedge \forall b \in \Sigma, LS(b) \cap {}^\bullet\ell = \emptyset \Rightarrow s(b) = s'(b).$$

- (or 1-safe Petri nets with mutually exclusive places)

Finite Automata Networks

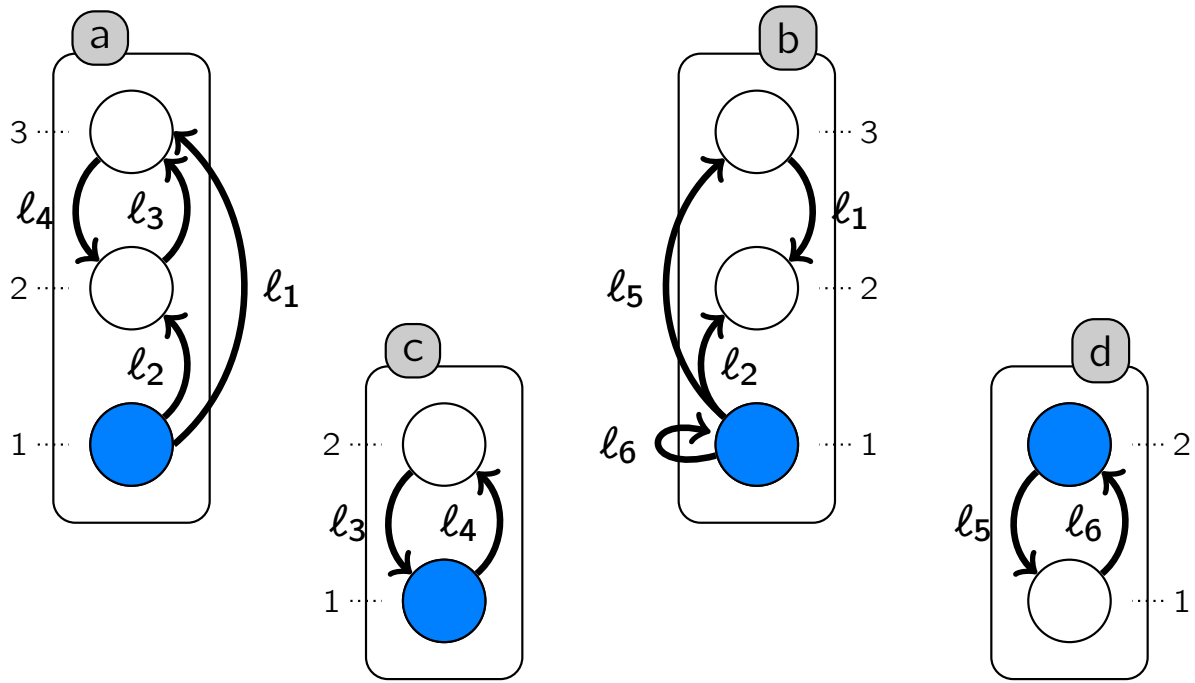


- transition ℓ pre-condition: ${}^\bullet\ell = \{a_i \mid a_i \xrightarrow{\ell} a_j\}$:

$$s \to s' \overset{\Delta}{\Leftrightarrow} \exists \ell \in \mathcal{L} : \forall a_i \in {}^\bullet\ell, s(a) = a_i \wedge \forall a_j \in \ell^\bullet, s'(a) = a_j$$
$$\wedge \forall b \in \Sigma, LS(b) \cap {}^\bullet\ell = \emptyset \Rightarrow s(b) = s'(b).$$

- (or 1-safe Petri nets with mutually exclusive places)

Finite Automata Networks

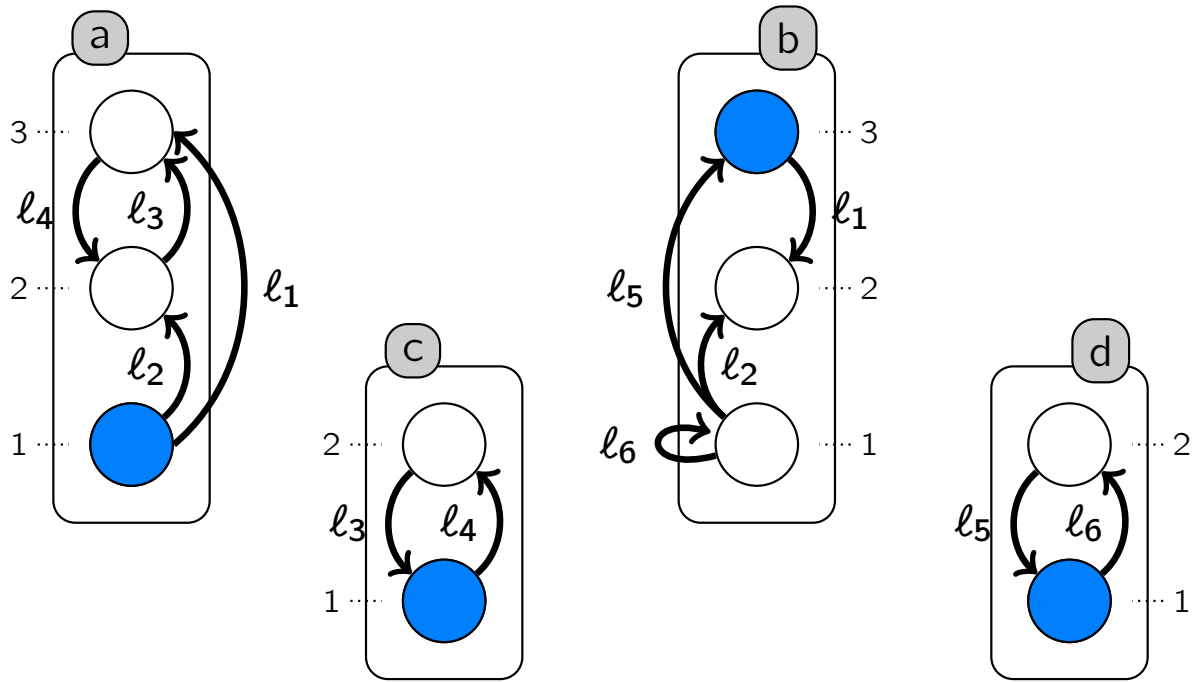


- transition ℓ pre-condition: ${}^\bullet\ell = \{a_i \mid a_i \xrightarrow{\ell} a_j\}$:

$$s \to s' \overset{\Delta}{\Leftrightarrow} \exists \ell \in \mathcal{L} : \forall a_i \in {}^\bullet\ell, s(a) = a_i \wedge \forall a_j \in \ell^\bullet, s'(a) = a_j$$
$$\wedge \forall b \in \Sigma, LS(b) \cap {}^\bullet\ell = \emptyset \Rightarrow s(b) = s'(b).$$

- (or 1-safe Petri nets with mutually exclusive places)

Finite Automata Networks

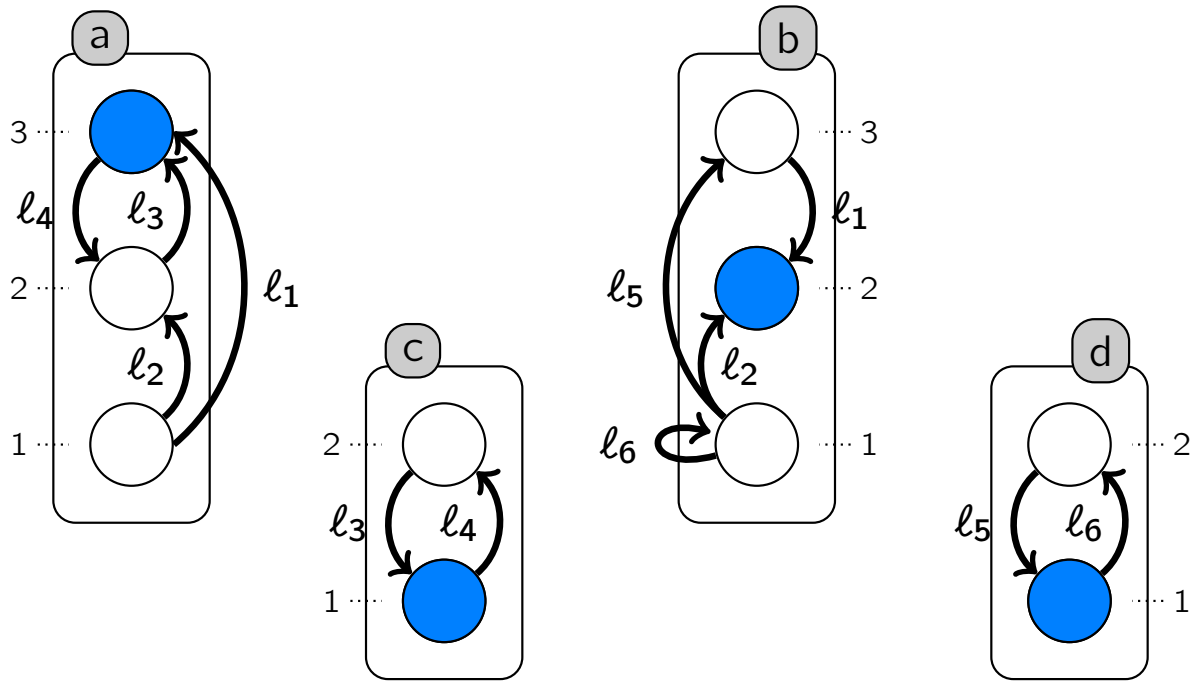


- transition ℓ pre-condition: ${}^\bullet\ell = \{a_i \mid a_i \xrightarrow{\ell} a_j\}$:

$$s \to s' \overset{\Delta}{\Leftrightarrow} \exists \ell \in \mathcal{L} : \forall a_i \in {}^\bullet\ell, s(a) = a_i \wedge \forall a_j \in \ell^\bullet, s'(a) = a_j$$
$$\wedge \forall b \in \Sigma, LS(b) \cap {}^\bullet\ell = \emptyset \Rightarrow s(b) = s'(b).$$

- (or 1-safe Petri nets with mutually exclusive places)

Finite Automata Networks

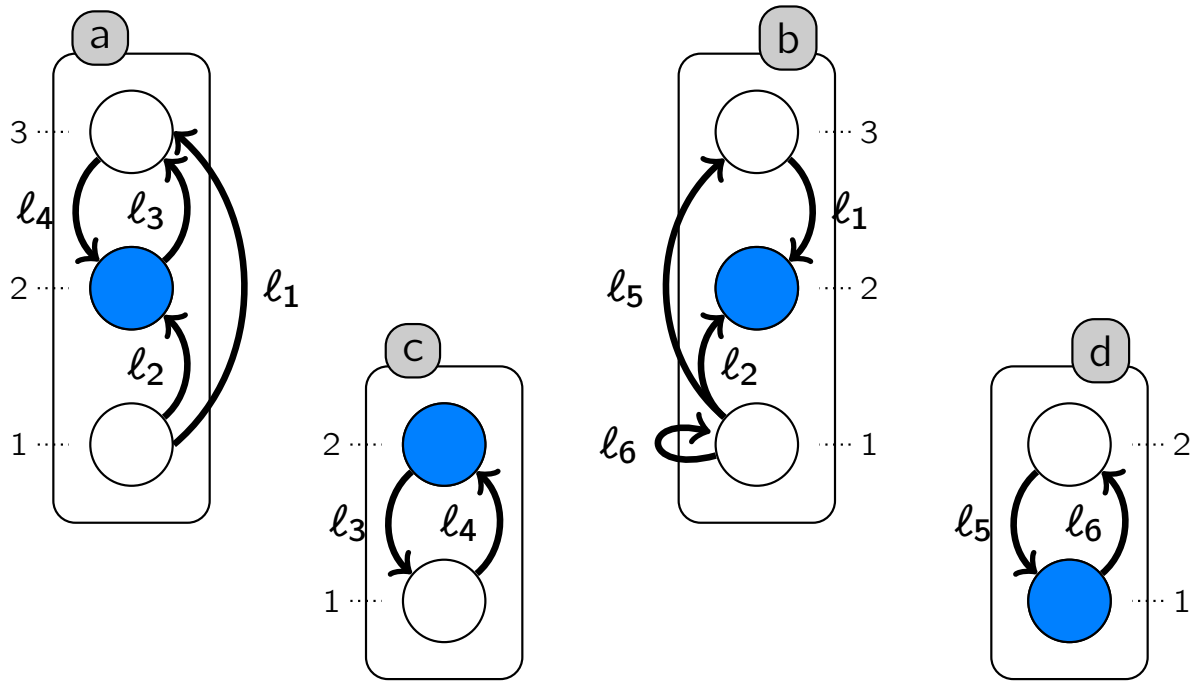


- transition ℓ pre-condition: ${}^\bullet\ell = \{a_i \mid a_i \xrightarrow{\ell} a_j\}$:

$$s \to s' \overset{\Delta}{\Leftrightarrow} \exists \ell \in \mathcal{L} : \forall a_i \in {}^\bullet\ell, s(a) = a_i \wedge \forall a_j \in \ell^\bullet, s'(a) = a_j$$
$$\wedge \forall b \in \Sigma, LS(b) \cap {}^\bullet\ell = \emptyset \Rightarrow s(b) = s'(b).$$

- (or 1-safe Petri nets with mutually exclusive places)

Finite Automata Networks



- transition ℓ pre-condition: ${}^\bullet\ell = \{a_i \mid a_i \xrightarrow{\ell} a_j\}$:

$$s \to s' \overset{\Delta}{\Leftrightarrow} \exists \ell \in \mathcal{L} : \forall a_i \in {}^\bullet\ell, s(a) = a_i \wedge \forall a_j \in \ell^\bullet, s'(a) = a_j$$
$$\wedge \forall b \in \Sigma, LS(b) \cap {}^\bullet\ell = \emptyset \Rightarrow s(b) = s'(b).$$

- (or 1-safe Petri nets with mutually exclusive places)

Motivation & Contribution

Settings

- Large number of automata;
- all different;
- small number of local states per automaton.

Capture dynamics

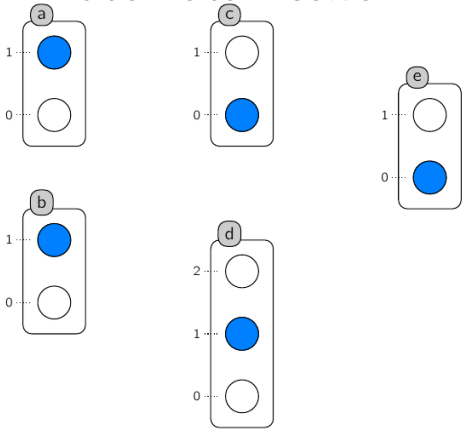
- Reachability.
- Cut sets for reachability (local states to kill).
- Attractors.

Abstract interpretation for traces of ANs

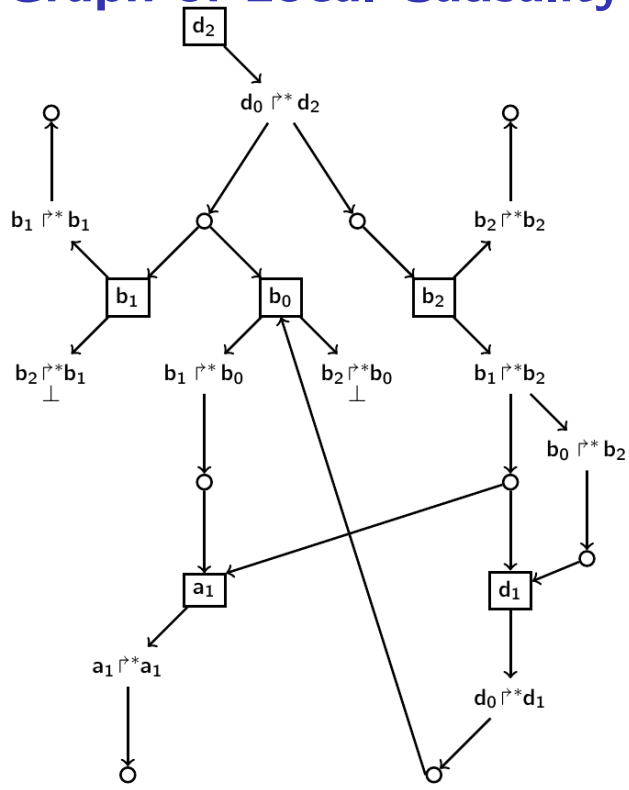
- Compact representation of possible traces.
- Over- and under-approximation of reachability + cut sets.
- Make tractable the analysis of very large application cases.

Overview

Automata network



Graph of Local Causality

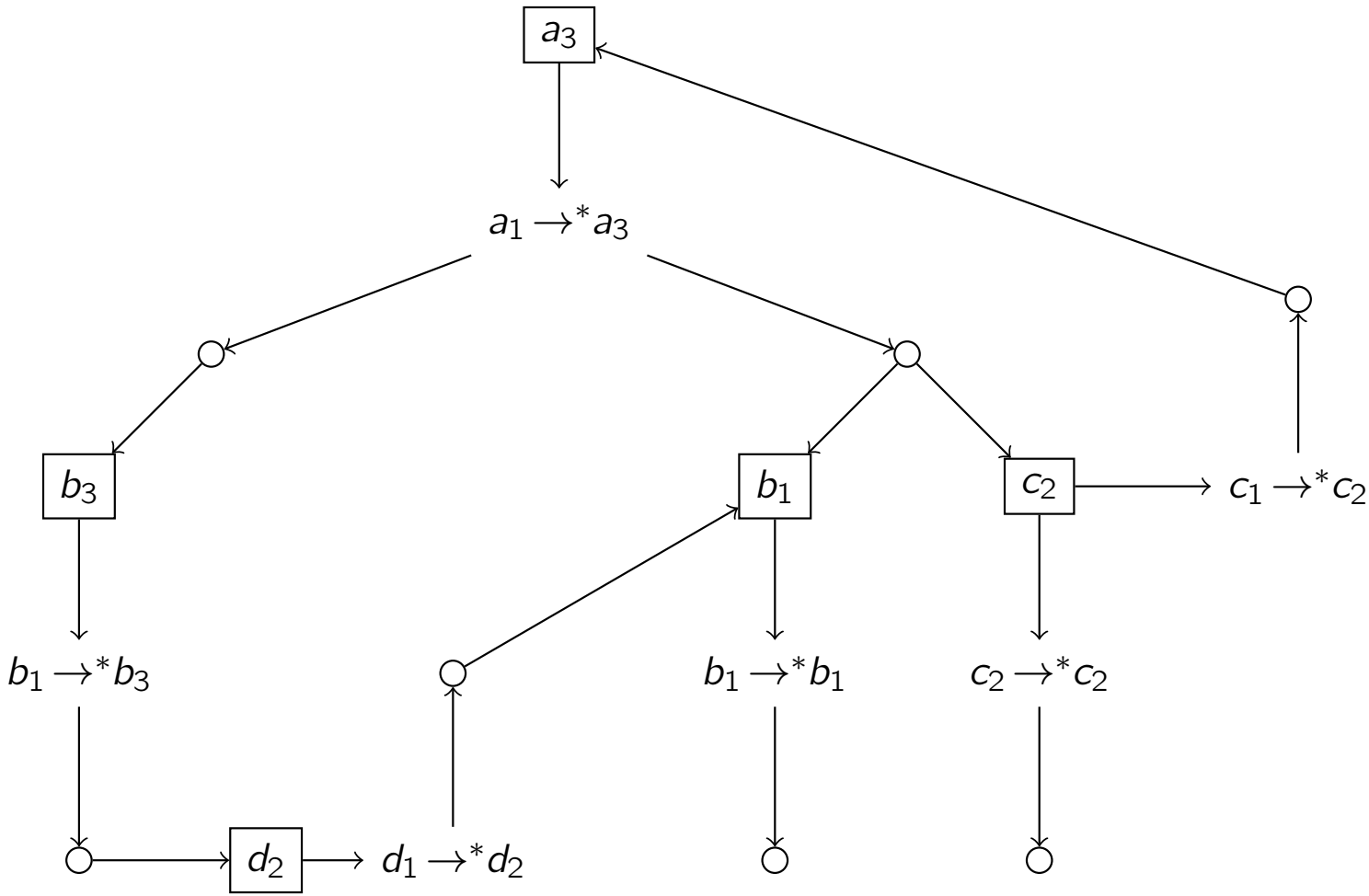


Over-approximation of reachability
Under-approximation of reachability
Under-approximation of cut sets

[Paulevé et al. in Math. Struct. in Comp. Sci. 2012]
[Paulevé et al. at CAV 2013]

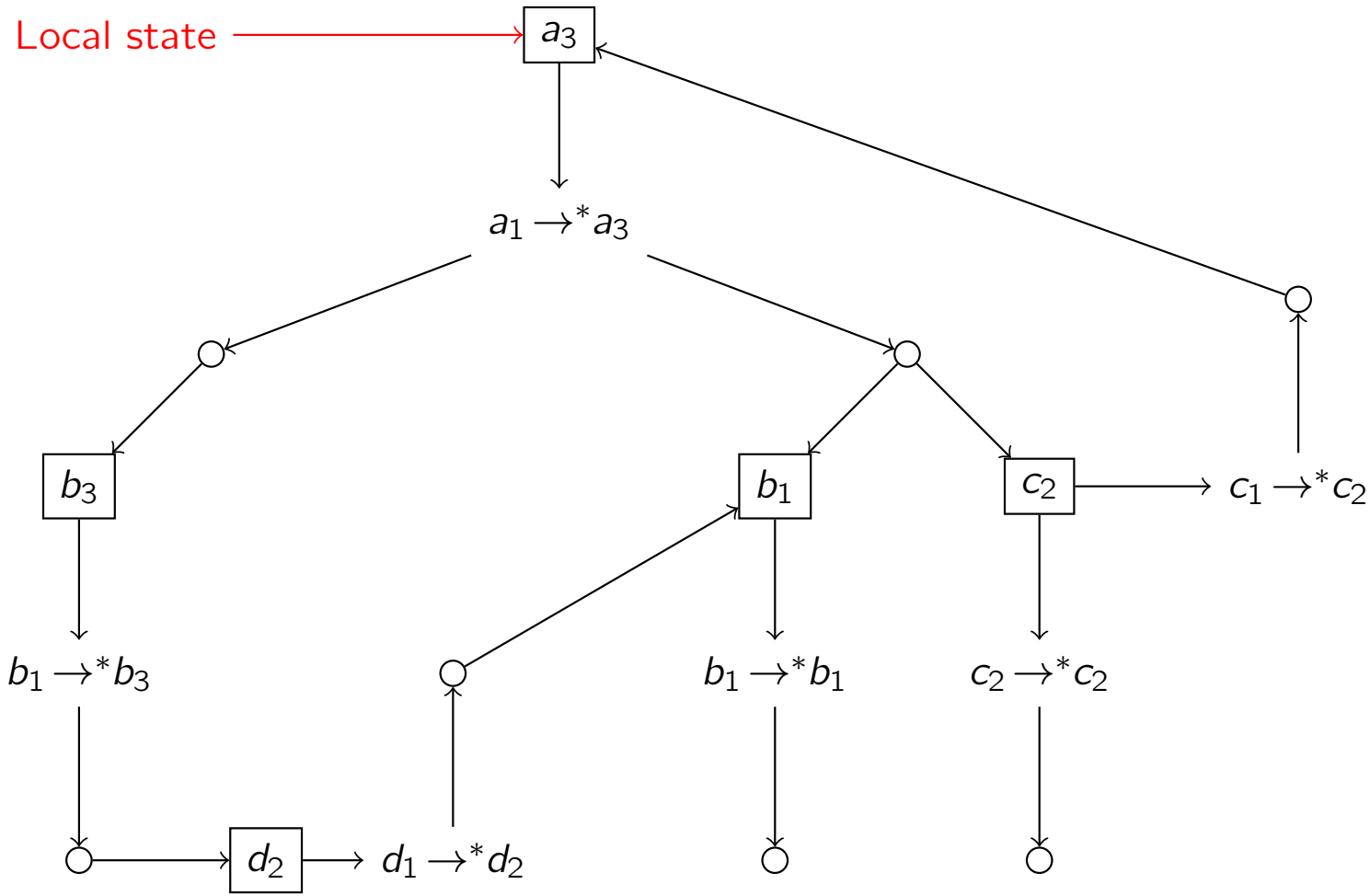
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{2\}\}$.



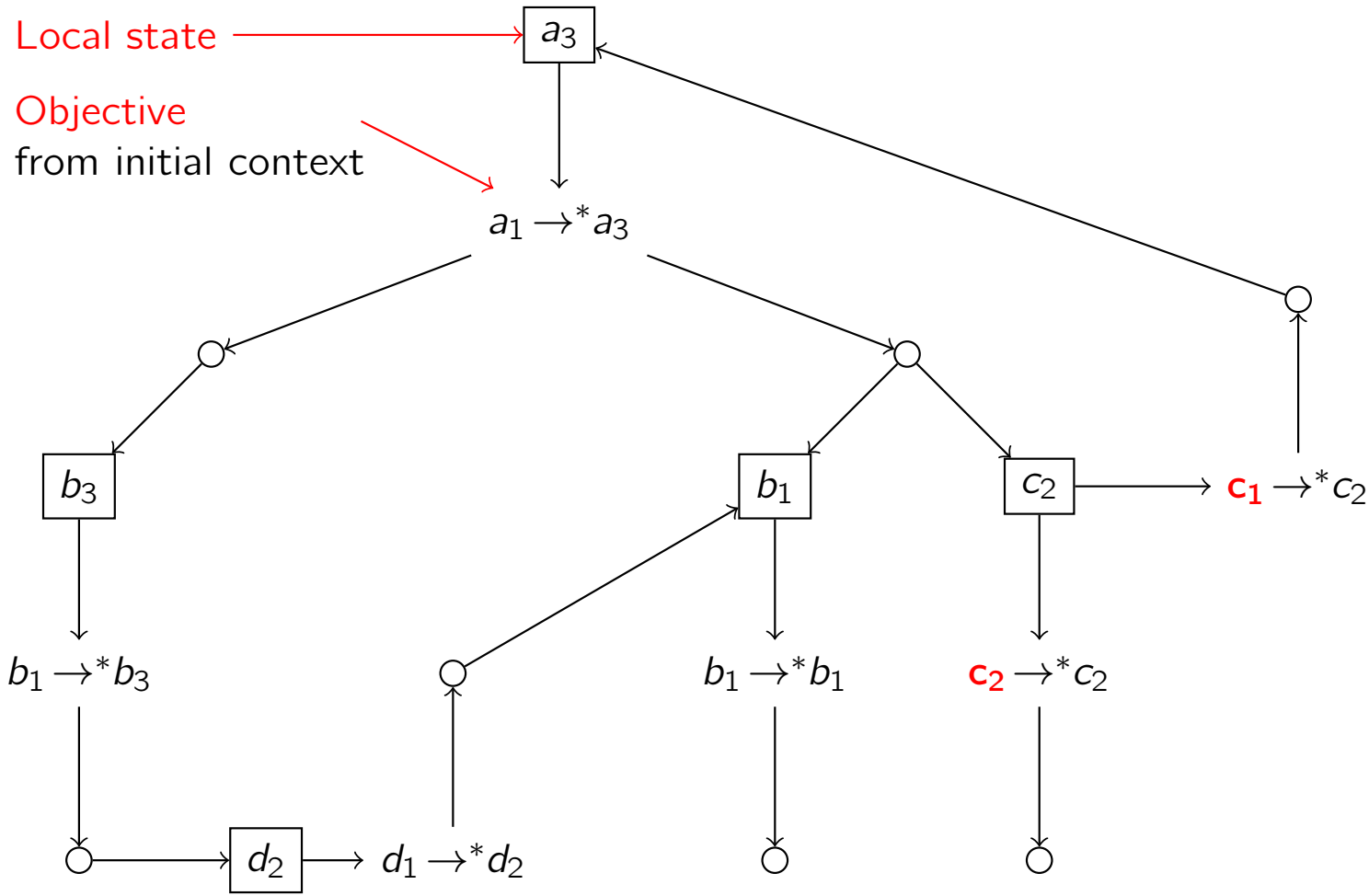
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{2\}\}$.



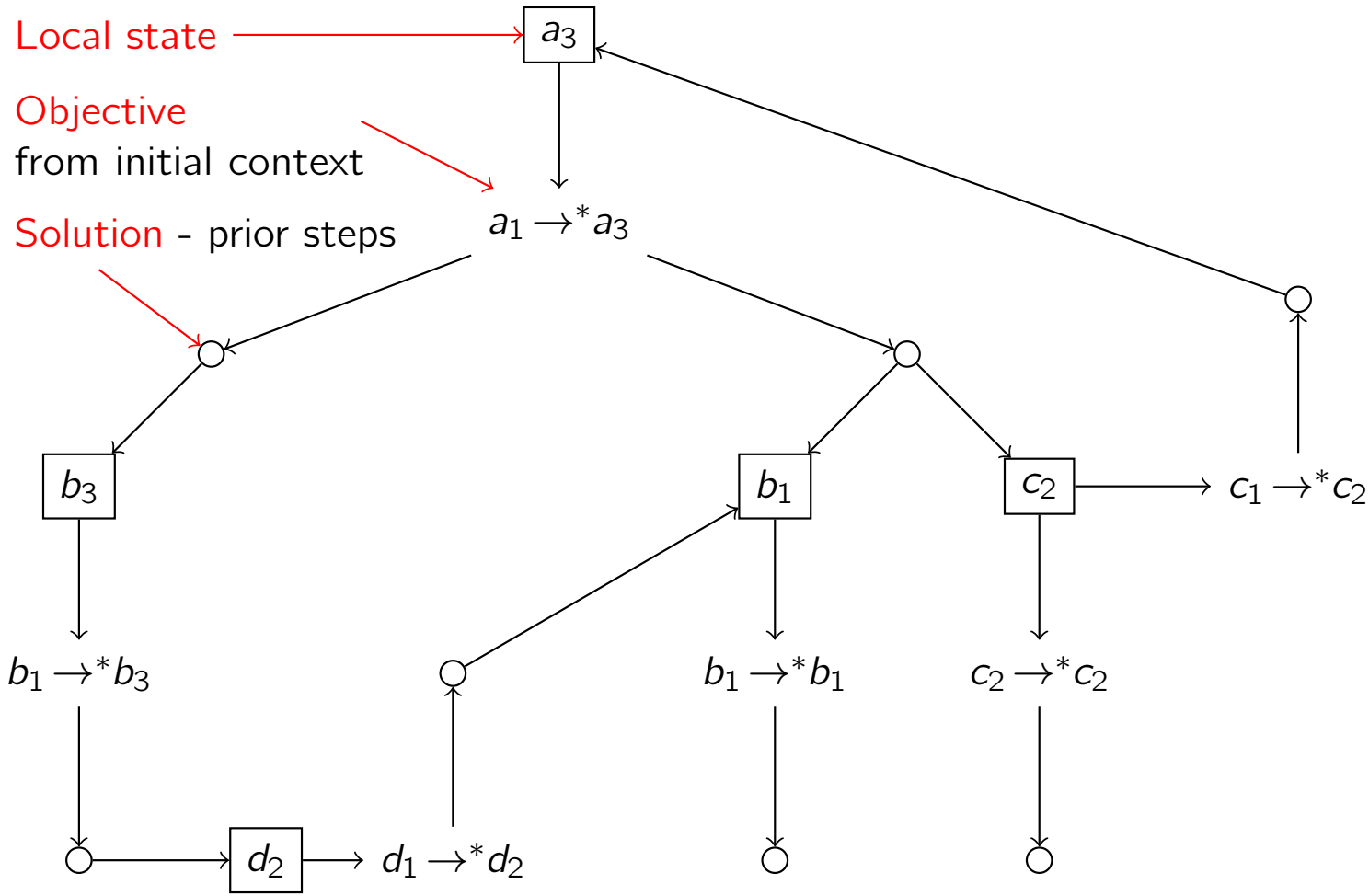
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{\mathbf{1}, \mathbf{2}\}; d \mapsto \{2\}\}$.



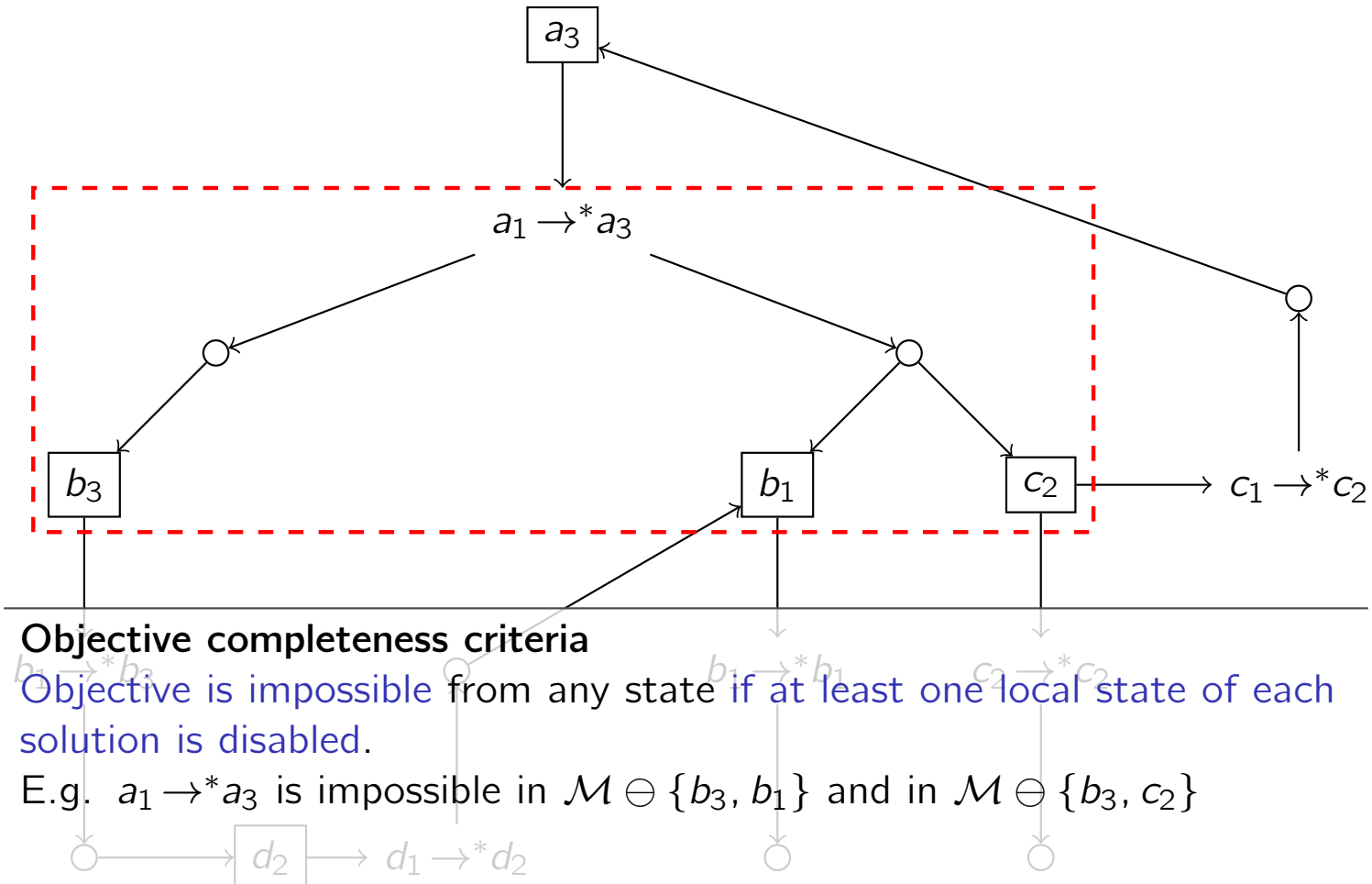
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{2\}\}$.



Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1,2\}; d \mapsto \{2\}\}$.

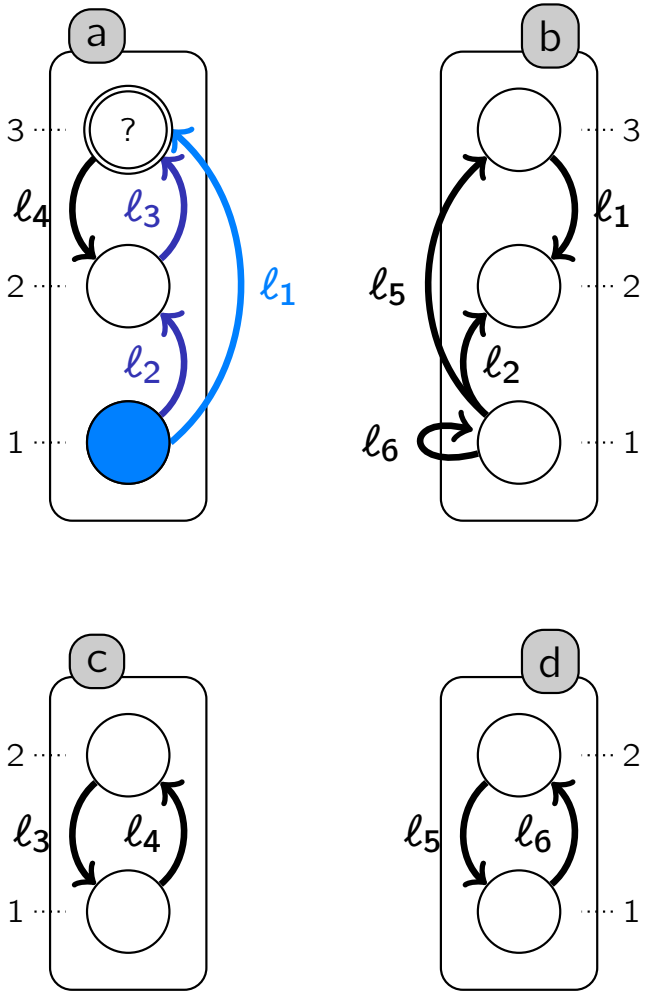


Objective completeness criteria

Objective is impossible from any state if at least one local state of each solution is disabled.

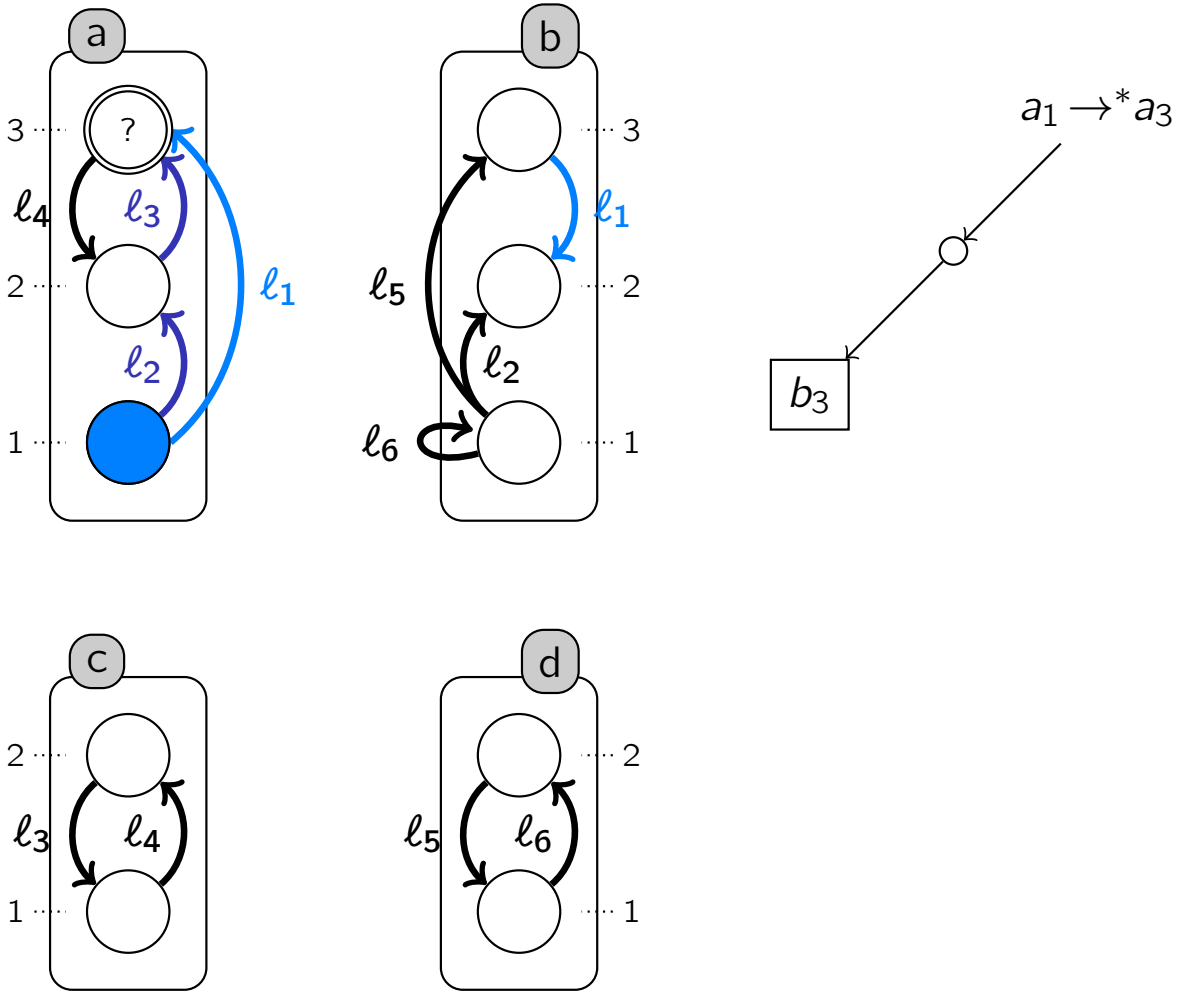
E.g. $a_1 \rightarrow^* a_3$ is impossible in $\mathcal{M} \ominus \{b_3, b_1\}$ and in $\mathcal{M} \ominus \{b_3, c_2\}$

Computing GLC for Automata Networks

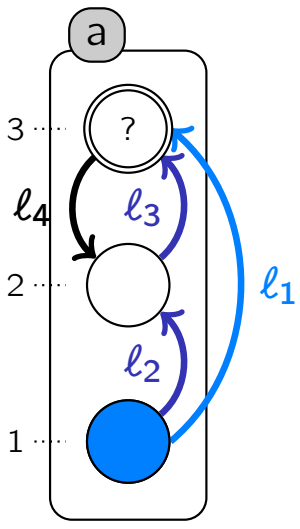


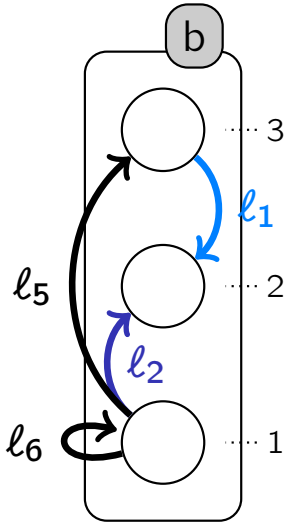
$a_1 \rightarrow^* a_3$

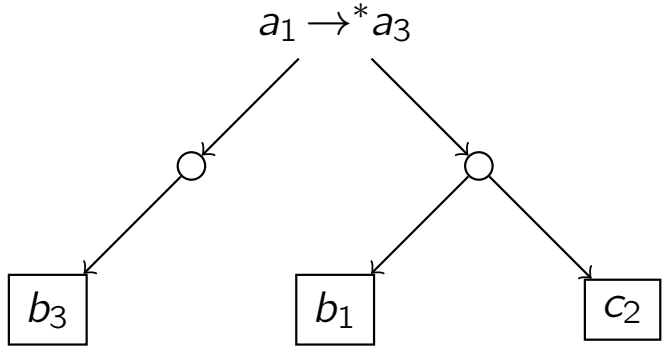
Computing GLC for Automata Networks



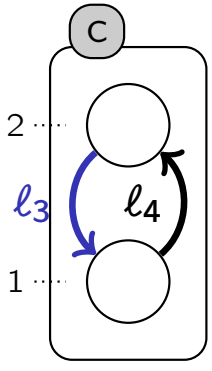
Computing GLC for Automata Networks

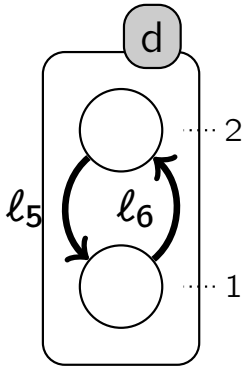




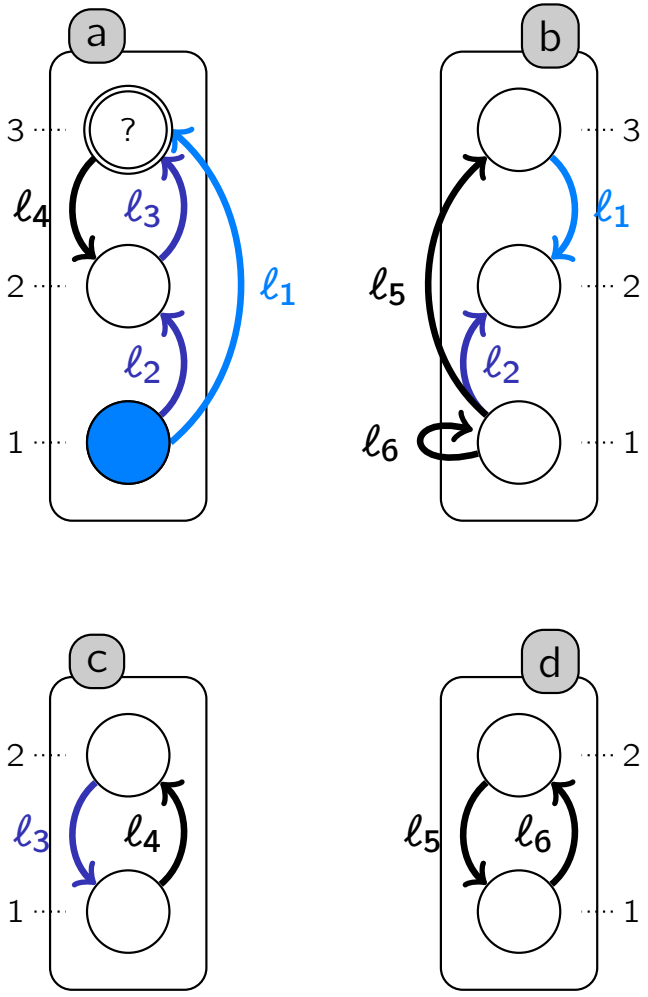


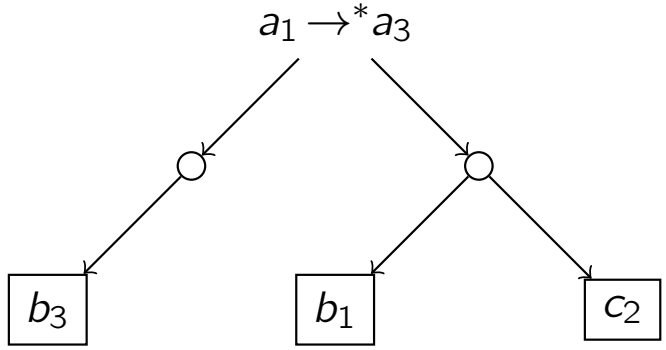
(ignore order, count, synchronism)





Computing GLC for Automata Networks





(ignore order, count, synchronism)

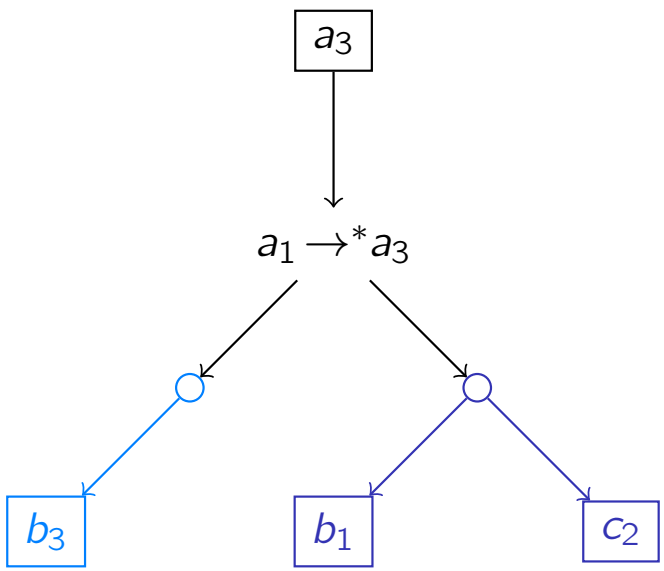
Complexity of GLC (construction + size of GLC)

- polynomial in the total number of local states;
- exponential in the number of local transitions within one automaton.

Abstract Interpretation with GLC

- ω: successive local reachability property: e.g. $a_i :: b_j :: \cdots z_l$.
- ς: initial context: e.g. $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1,2\}; d \mapsto \{2\}\}$.

$$\gamma_\varsigma(\omega) = \{\delta \in \textbf{traces} \mid \delta \text{ concretizes } \omega \wedge \delta \text{ starts in } \varsigma\}$$



$\Longrightarrow$

$$\gamma_\varsigma(a_3) = \gamma_\varsigma(b_3 :: a_3) \cup \gamma_\varsigma(b_1 :: c_2 :: a_3) \cup \gamma_\varsigma(c_2 :: b_1 :: a_3)$$

Reachability Analysis using GLC

Given

- ω: successive local reachability property: e.g. $a_i :: b_j :: \cdots z_l$.
- ς: initial context: e.g. $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{2\}\}$.

decide if $\gamma_\varsigma(\omega) \neq \emptyset$.

Results

- Necessary condition for general case.
- Stronger necessary conditions for Asynchronous ANs.
- Sufficient condition for Async. ANs with sync. arity ≤ 2.

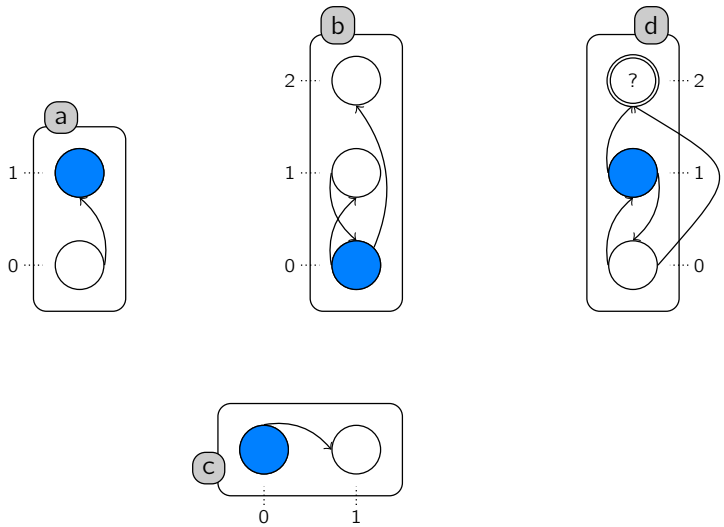
Over-: polynomial in the size of the GLC;
Under-: exp. with nb of solutions per objective.

Cut sets for a_i

- Sets of local states whose activity is necessary for $\gamma_\varsigma(a_i) \neq \emptyset$.
- Under-approximation using GLC (some are missed, some are non-minimal).
- General to any AN.

Over-approximation of Reachability

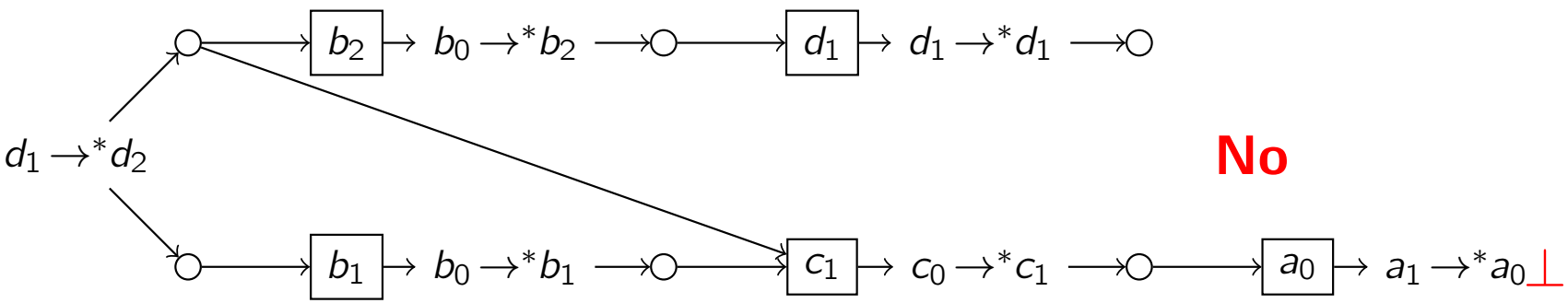
Example



Necessary condition for $\gamma_s(d_2) \neq \emptyset$:

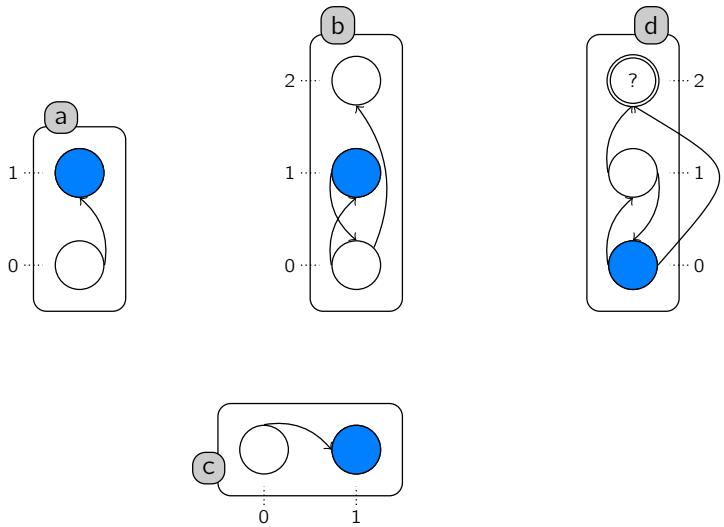
There exists a traversal of the GLC s.t.:

- objective $\rightarrow$ follow at least one solution;
- local state $\rightarrow$ follow all objectives;
- no cycle.



Over-approximation of Reachability

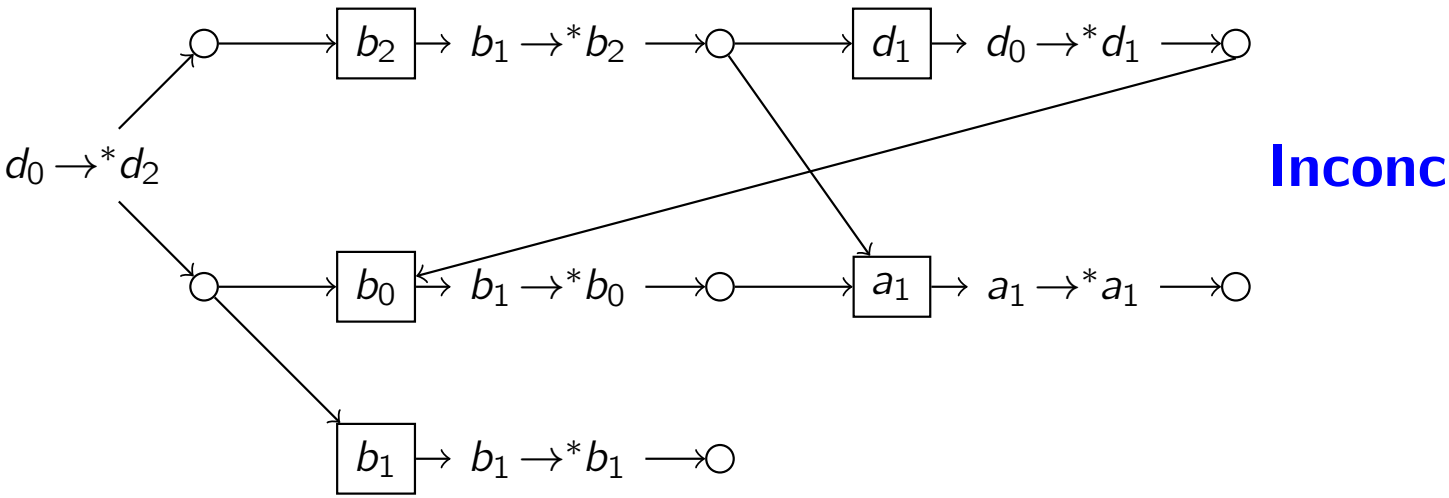
Example



Necessary condition for $\gamma_s(d_2) \neq \emptyset$:

There exists a traversal of the GLC s.t.:

- objective $\rightarrow$ follow at least one solution;
- local state $\rightarrow$ follow all objectives;
- no cycle.



Inconc

Over-approximation of Reachability

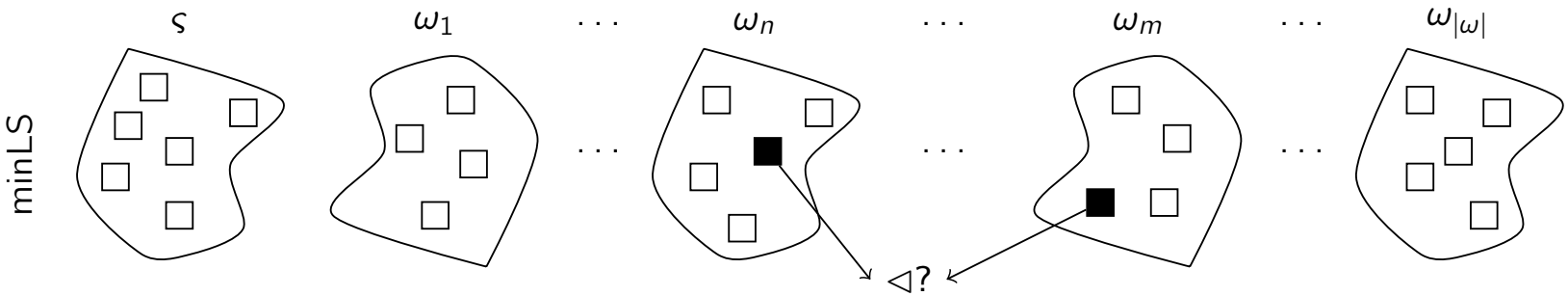
Stronger conditions

Scope: Asynchronous ANs (transitions change only one automaton).

Take sequentiality into account

- $\gamma_\varsigma(a_i :: \omega) \neq \emptyset \Longrightarrow \gamma_{max_\varsigma}(\omega) \neq \emptyset$
- Over-approximate next context using GLC.
- (time polynomial with size of GLC).

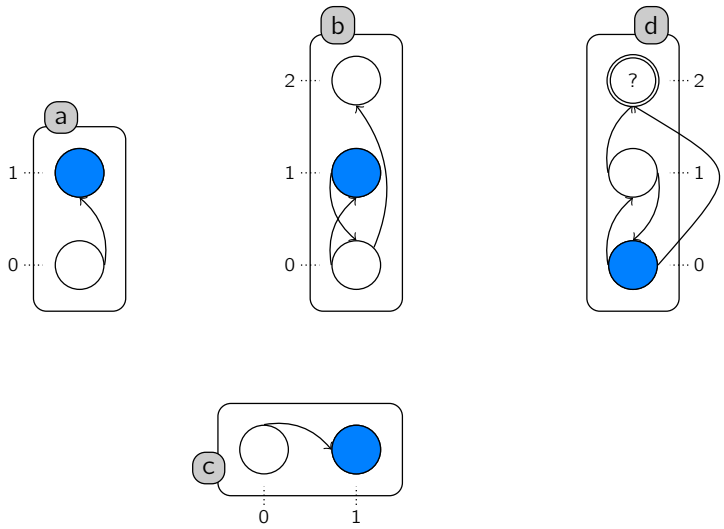
Local states occurrence order constraints



- Extract local states necessarily used for reachability using GLC.
- Check ordering constraints (e.g.: $a_j \lhd a_i$ if $\text{sol}(a_i \rightarrow^* a_j) = \emptyset$).

Under-approximation of Reachability

Example

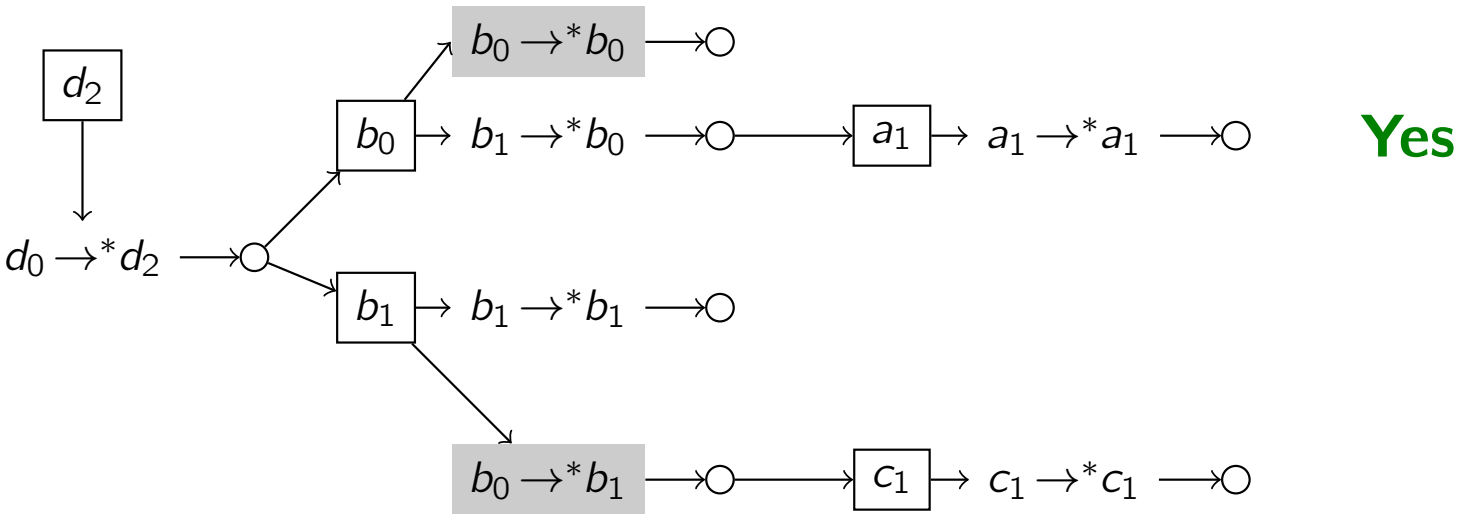


Scope

Asynchronous ANs with transition arity ≤ 2.

Sufficient condition for $\gamma_{\varsigma}(d_2) \neq \emptyset$:

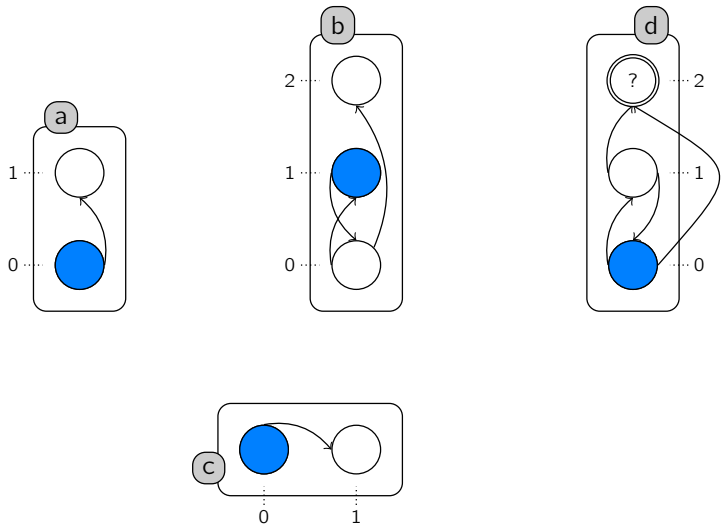
- GLC' has no cycle;
- each objective has at least one solution.



Yes

Under-approximation of Reachability

Example

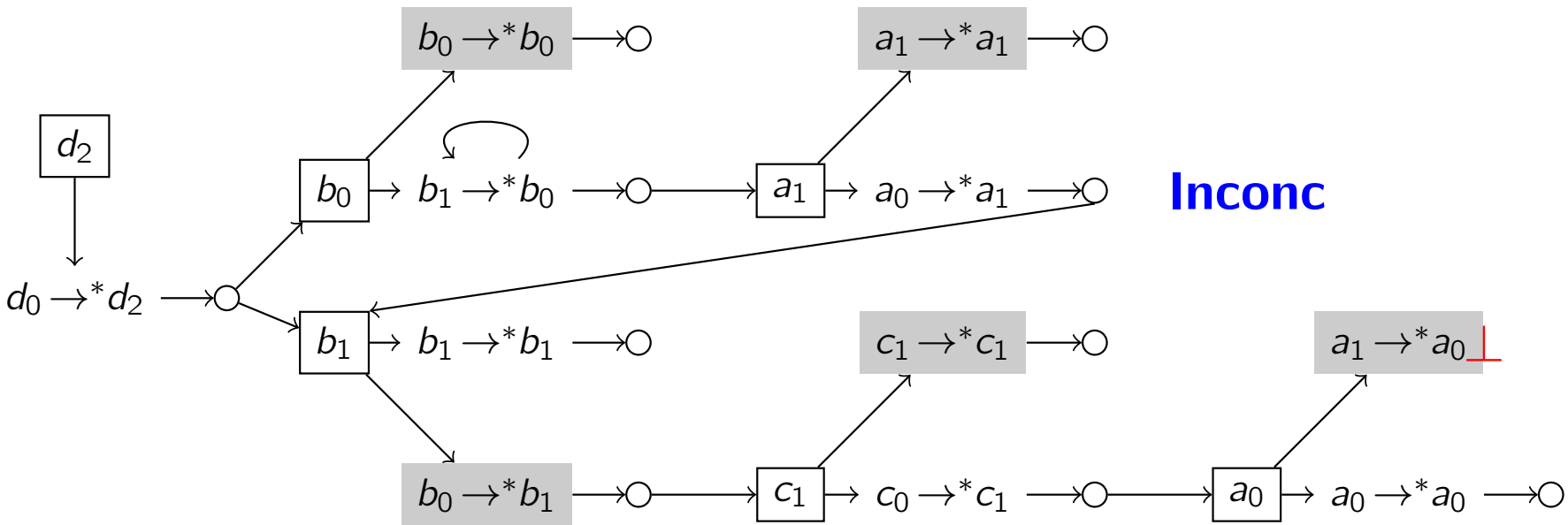


Scope

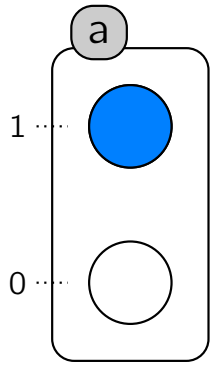
Asynchronous ANs with transition arity ≤ 2.

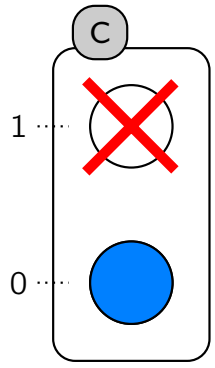
Sufficient condition for $\gamma_s(d_2) \neq \emptyset$:

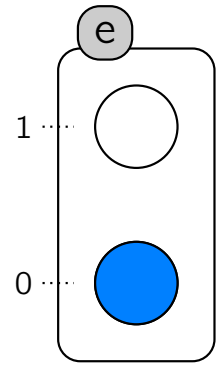
- GLC' has no cycle;
- each objective has at least one solution.



Cut Sets for Reachability

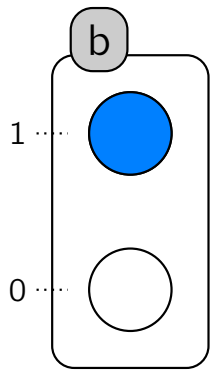


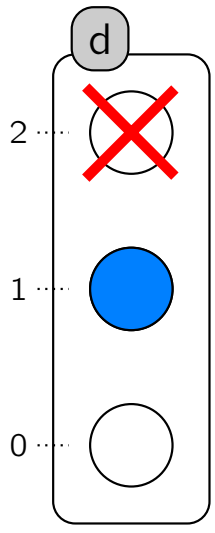


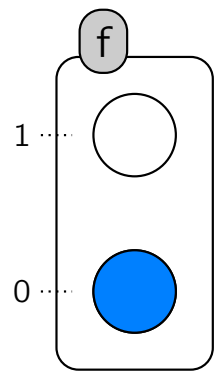


Set of **local states** that **if all disabled break reachability** from given initial states

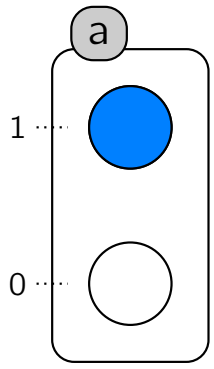
e.g. $\{c_1, d_2\}$

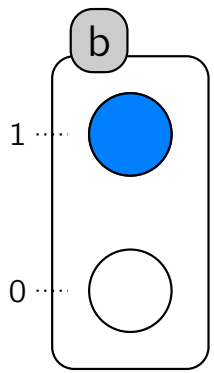


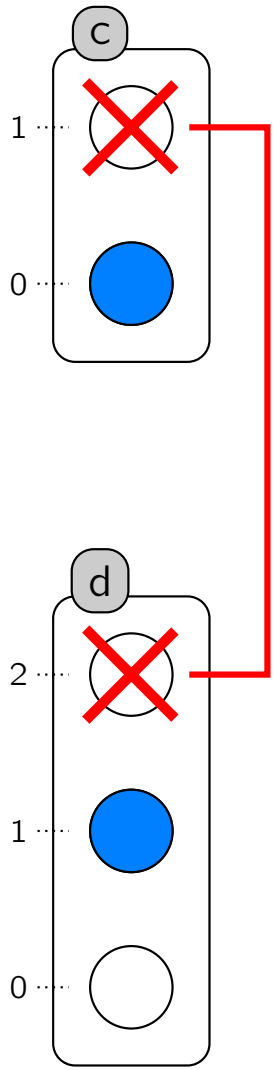


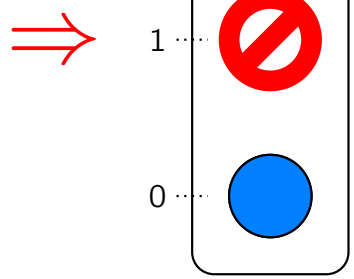


Cut Sets for Reachability



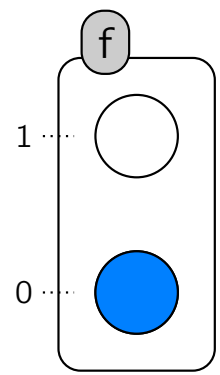






Set of **local states** that **if all disabled break reachability** from given initial states

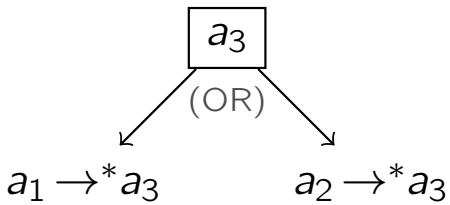
e.g. $\{c_1, d_2\}$



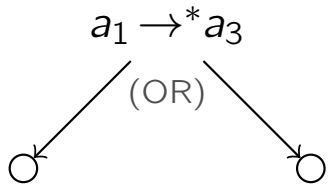
Cut Sets Under-Approximation

Associate to each node sets of local states intersecting *any* trace from given context.

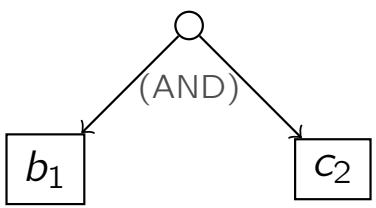
$$\mathbb{V} : \text{nodes} \mapsto \wp(\wp^{\leq N}(\mathcal{O}bs)), \mathcal{O}bs \subset \mathbf{LS}$$



$$\mathbb{V}(a_3) = \mathbb{V}(a_1 \rightarrow^* a_3) \tilde{\times} \mathbb{V}(a_2 \rightarrow^* a_3) \cup \{\{a_3\}\}$$



$$\mathbb{V}(a_1 \rightarrow^* a_3) = \mathbb{V}(sol^1) \tilde{\times} (sol^2)$$



$$\mathbb{V}(sol^1) = \mathbb{V}(b_1) \cup \mathbb{V}(c_2)$$

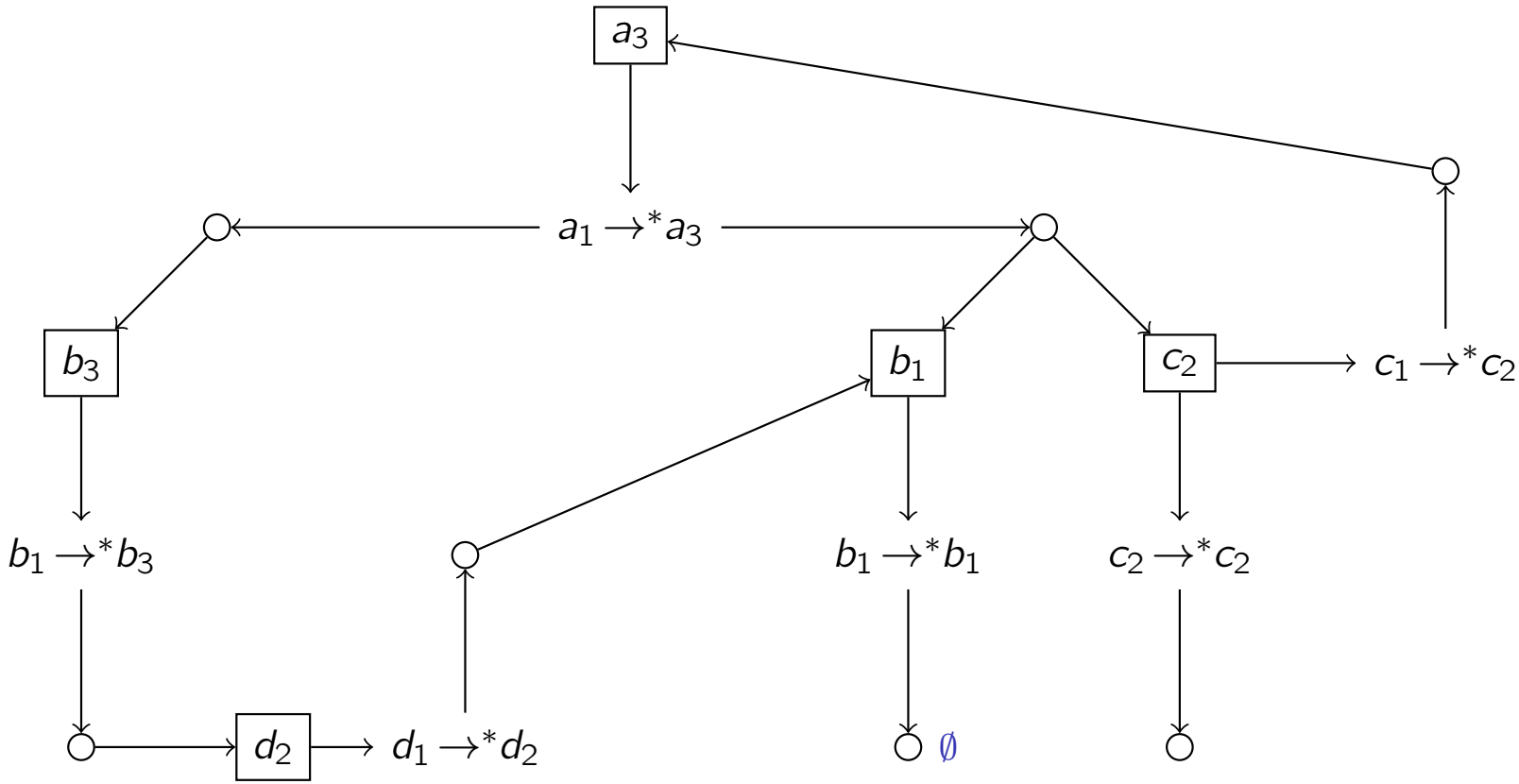
$$\{e^1, \ldots, e^n\} \tilde{\times} \{f^1, \ldots, f^m\} \triangleq \{e^i \cup f^j \mid i \in [1; n] \wedge j \in [1; m]\} \; ; \; e^i, f^j \in \wp^{\leq N}(\mathcal{O}bs)$$

Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

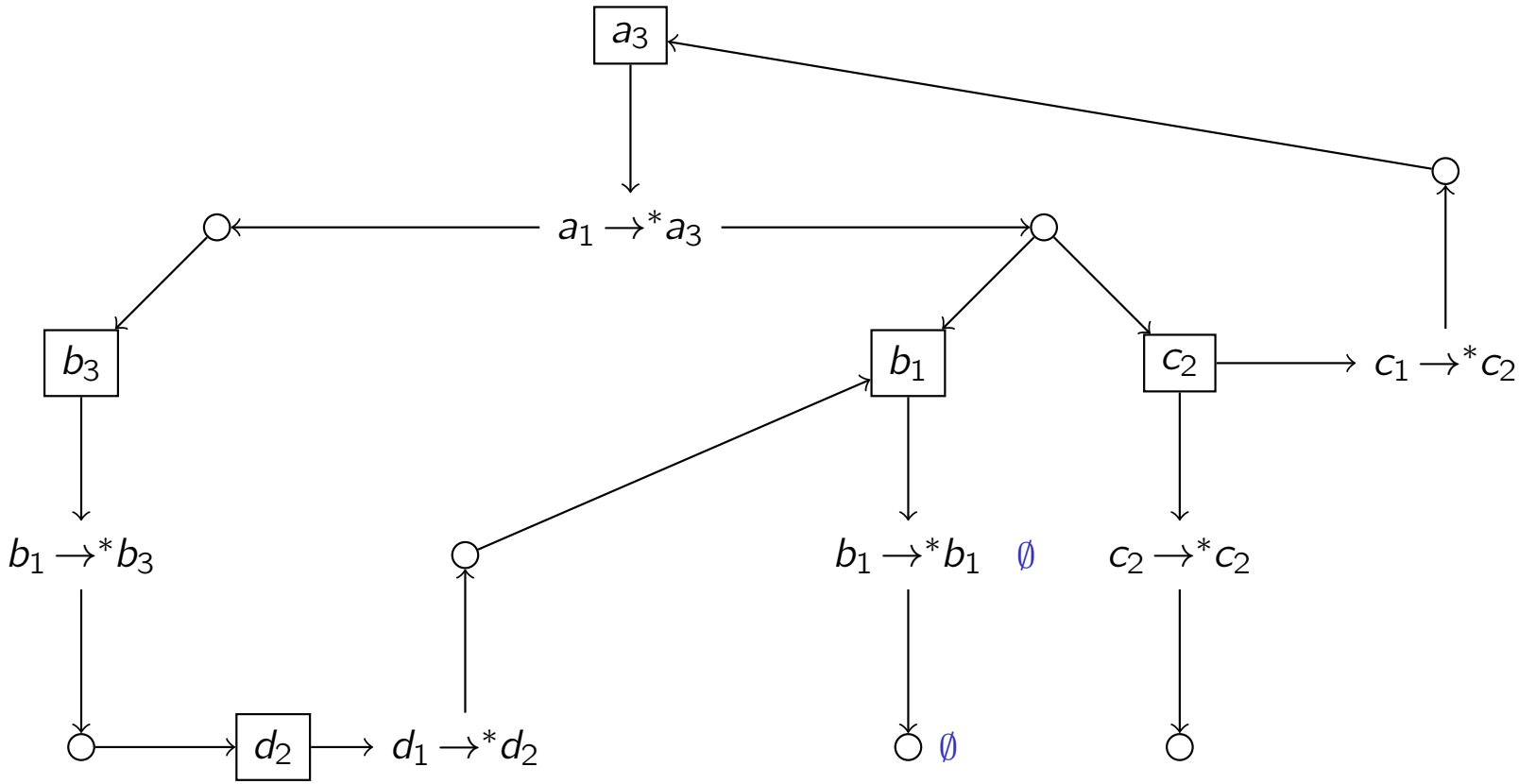


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

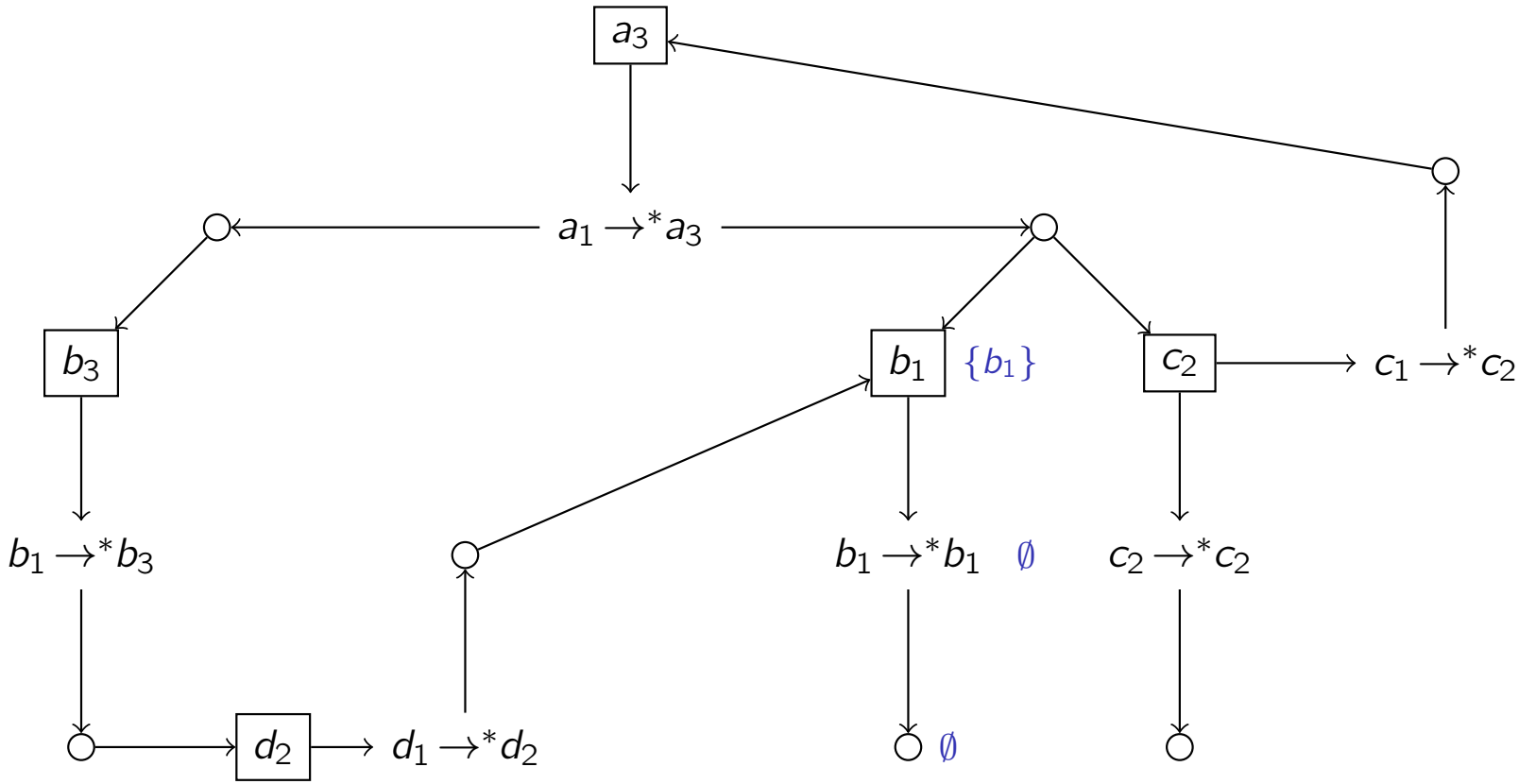


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

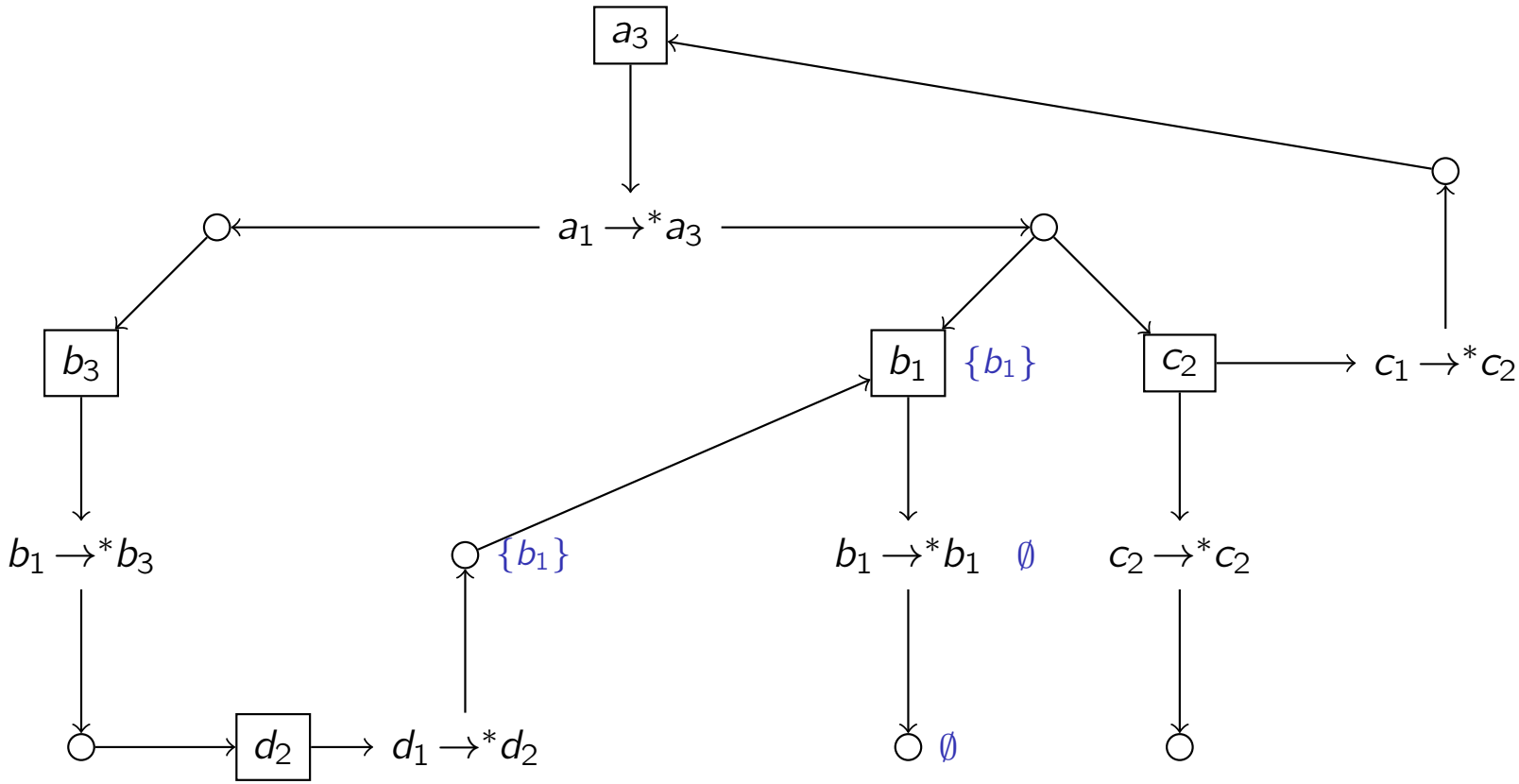


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

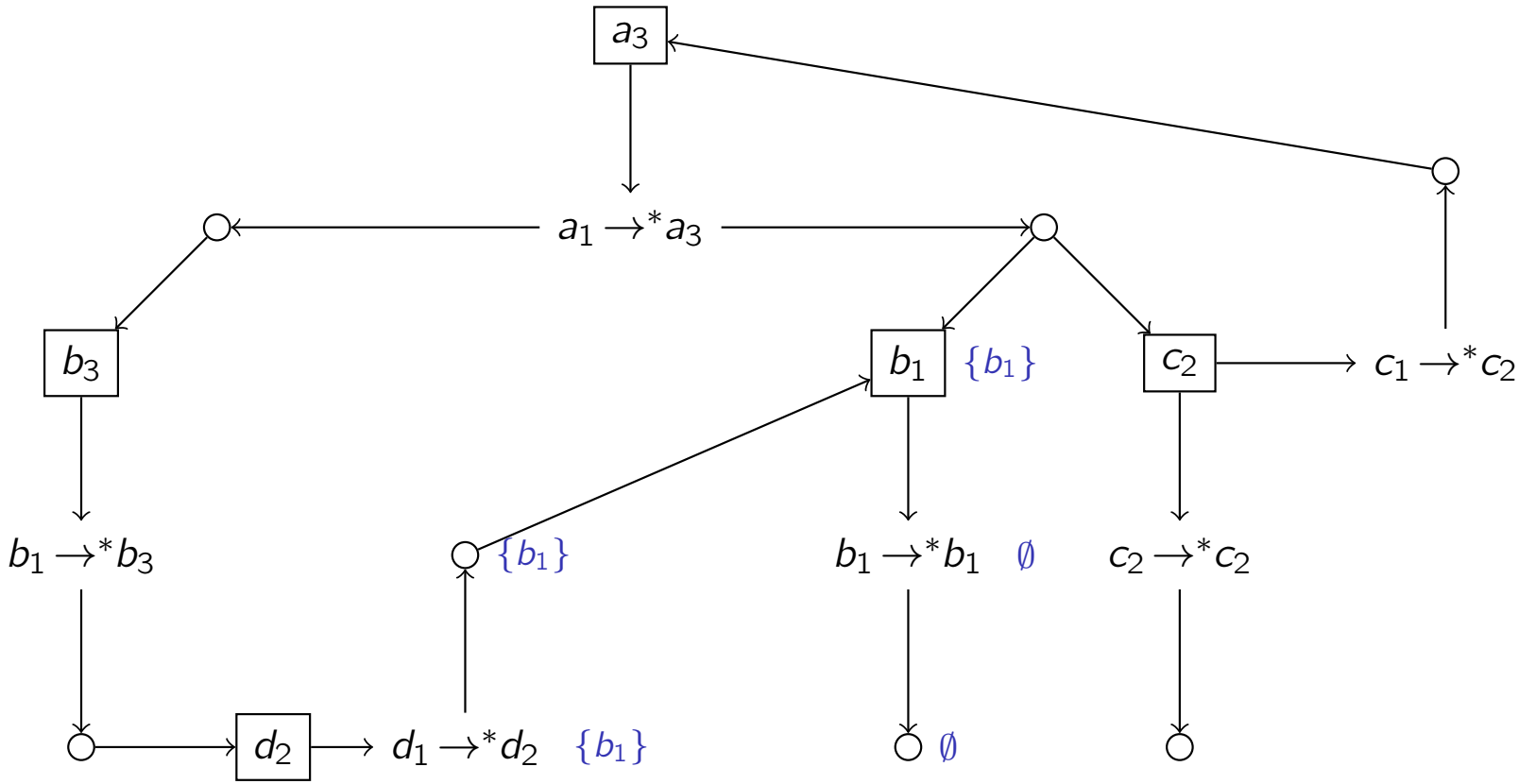


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

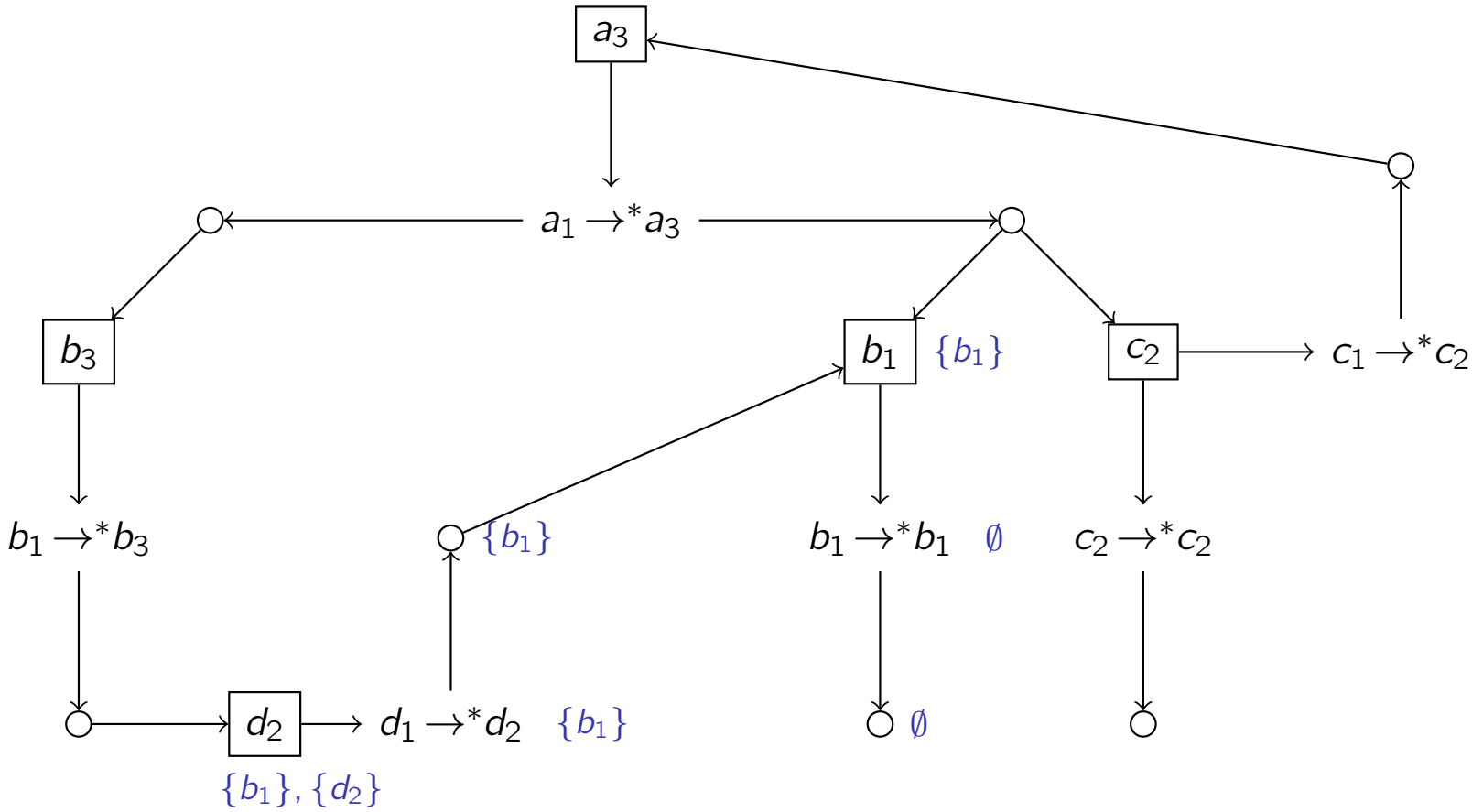


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

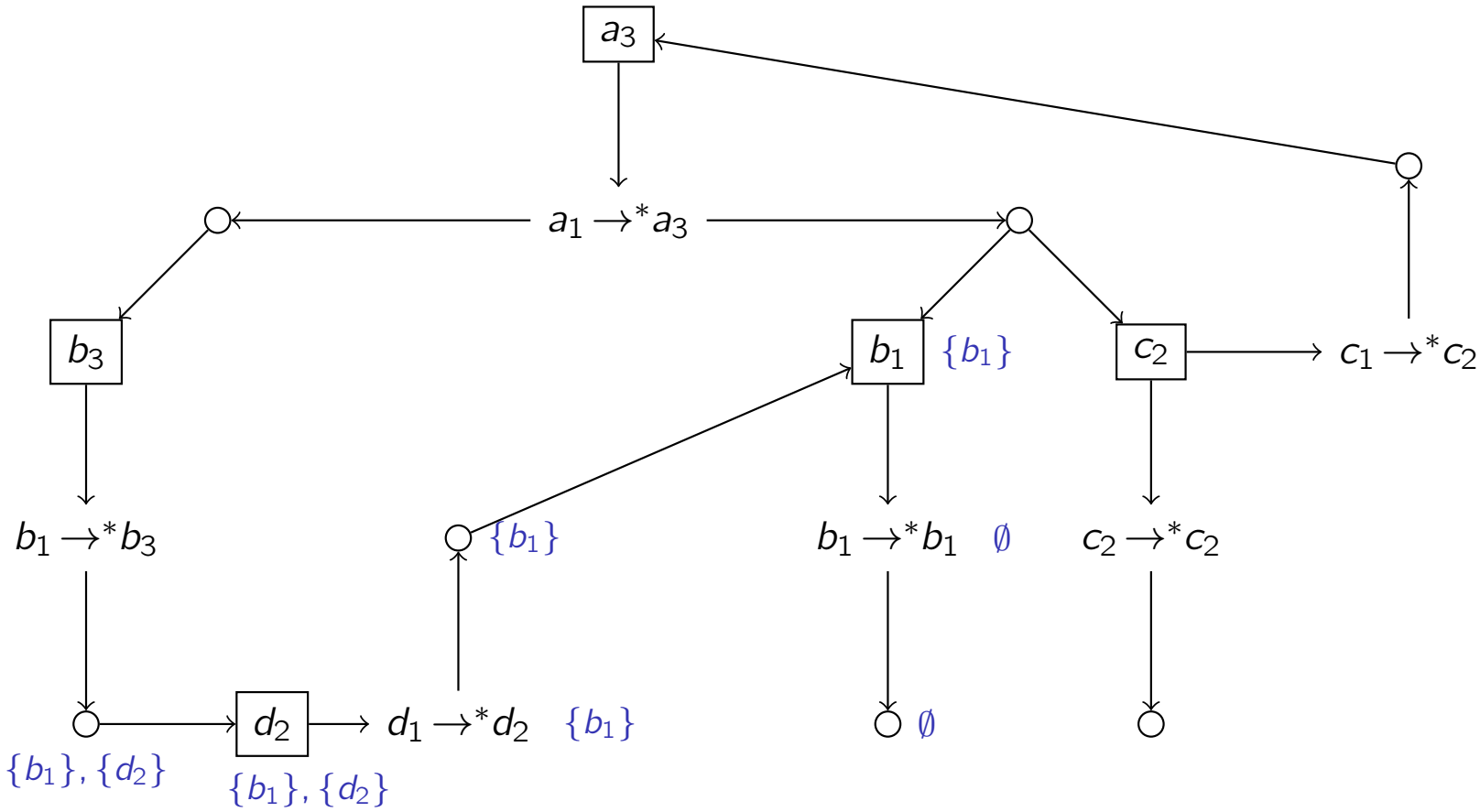


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

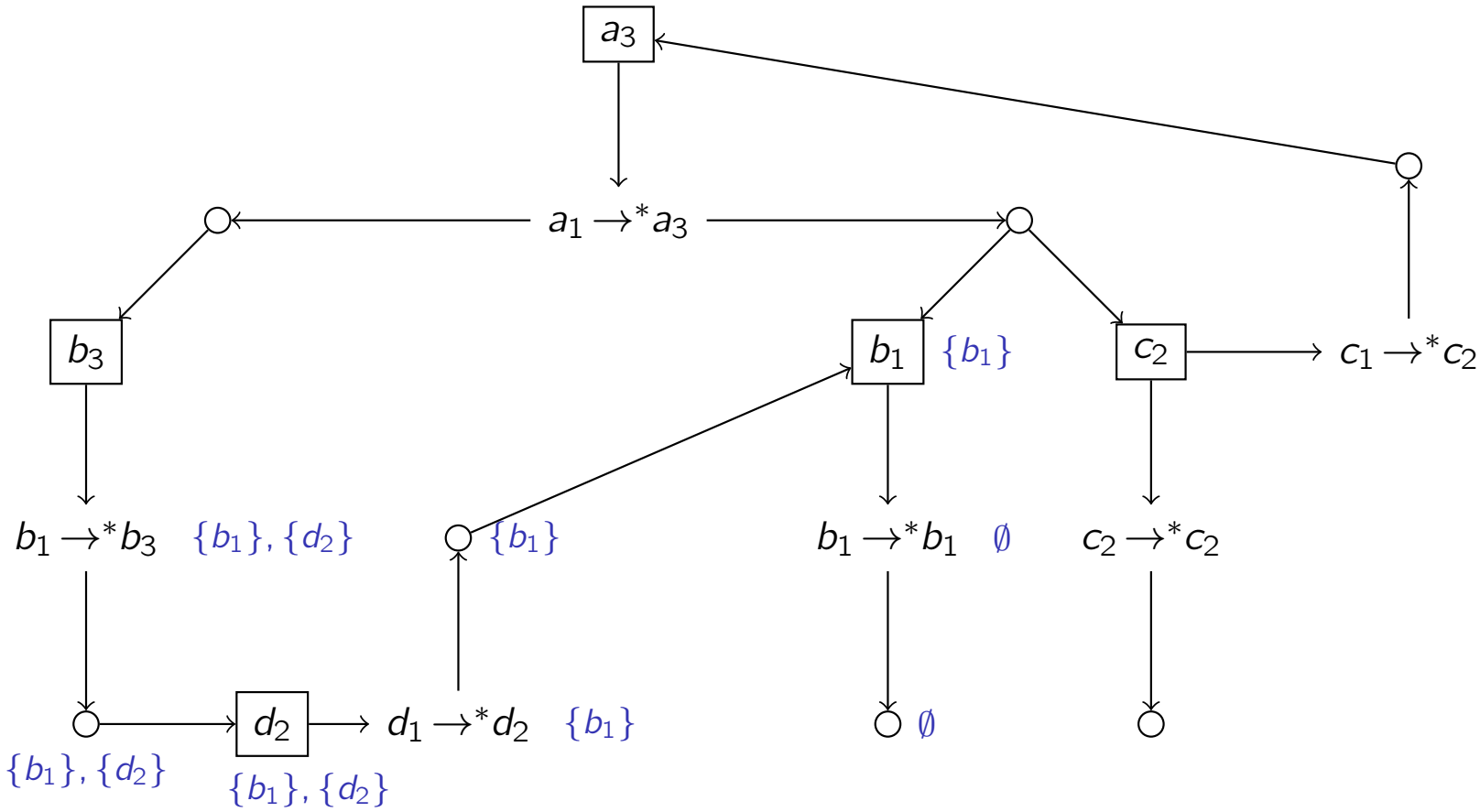


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

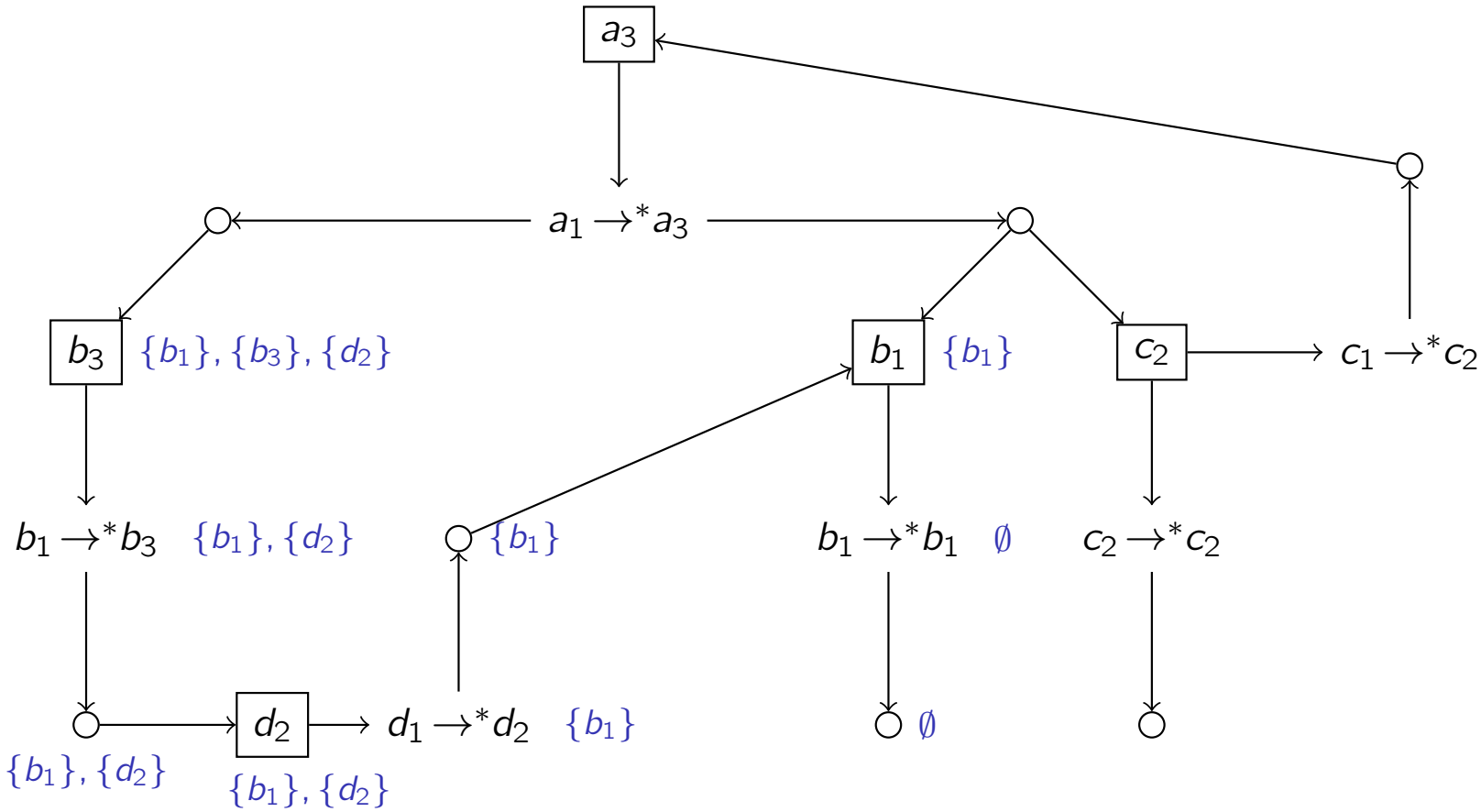


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

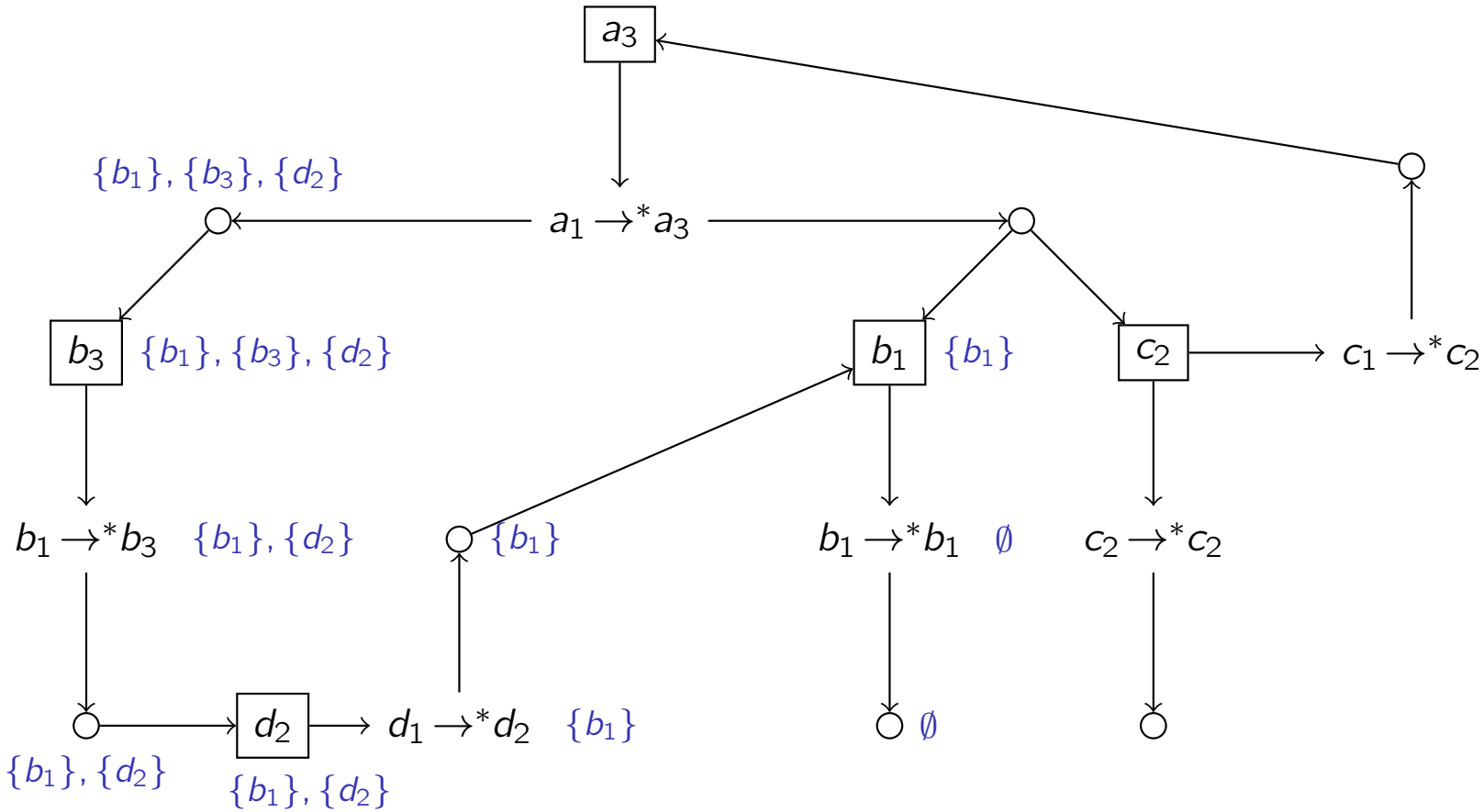


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

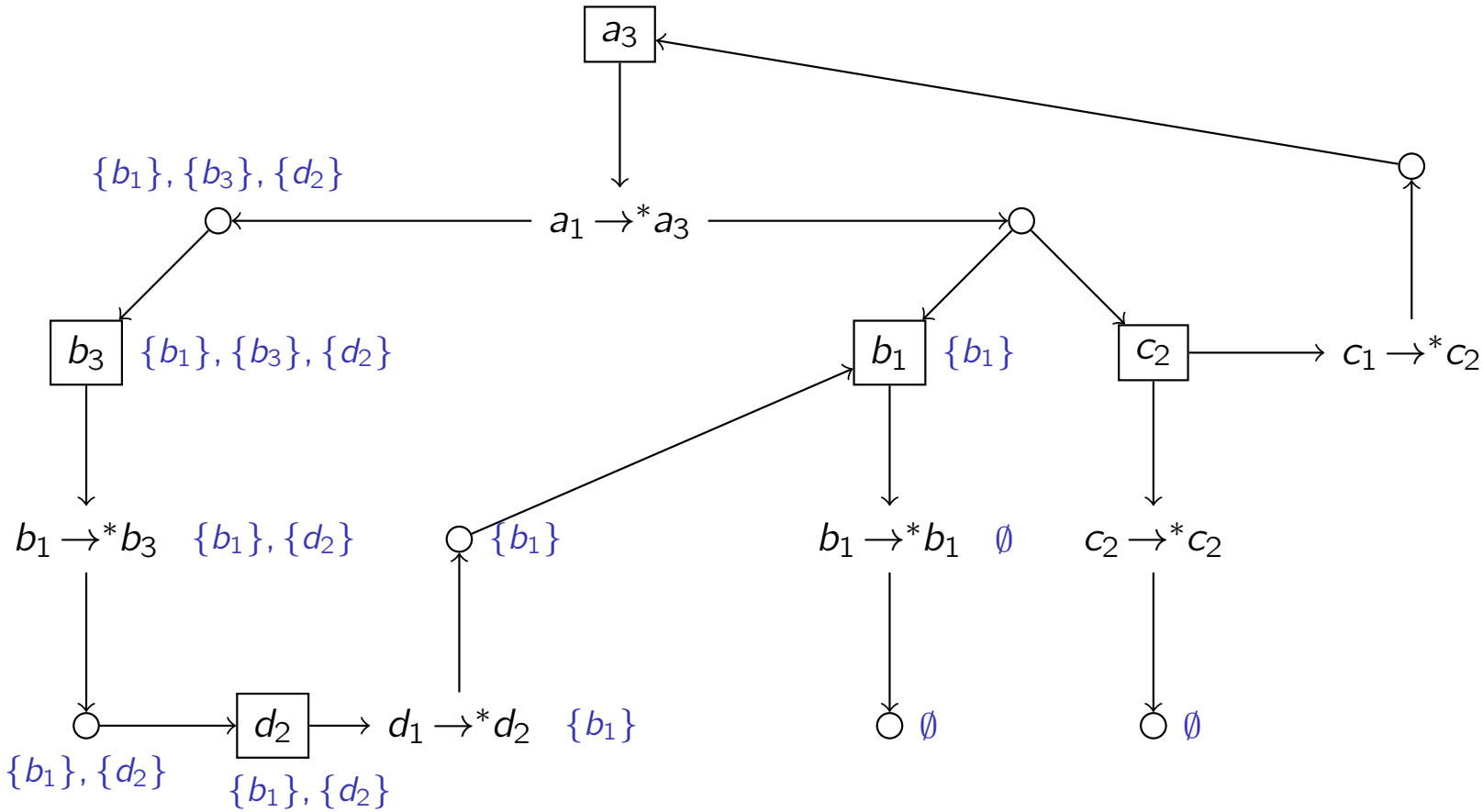


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

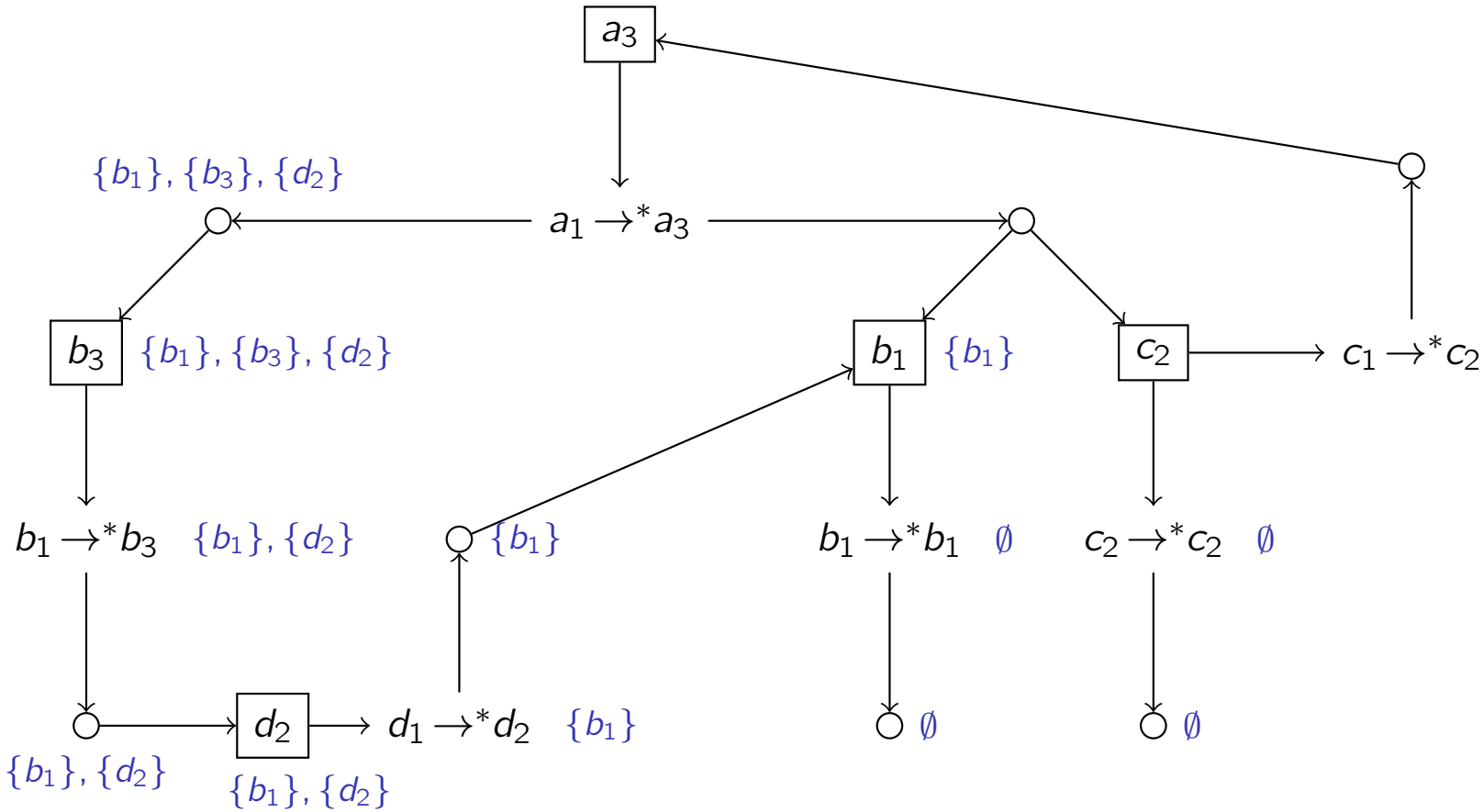


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

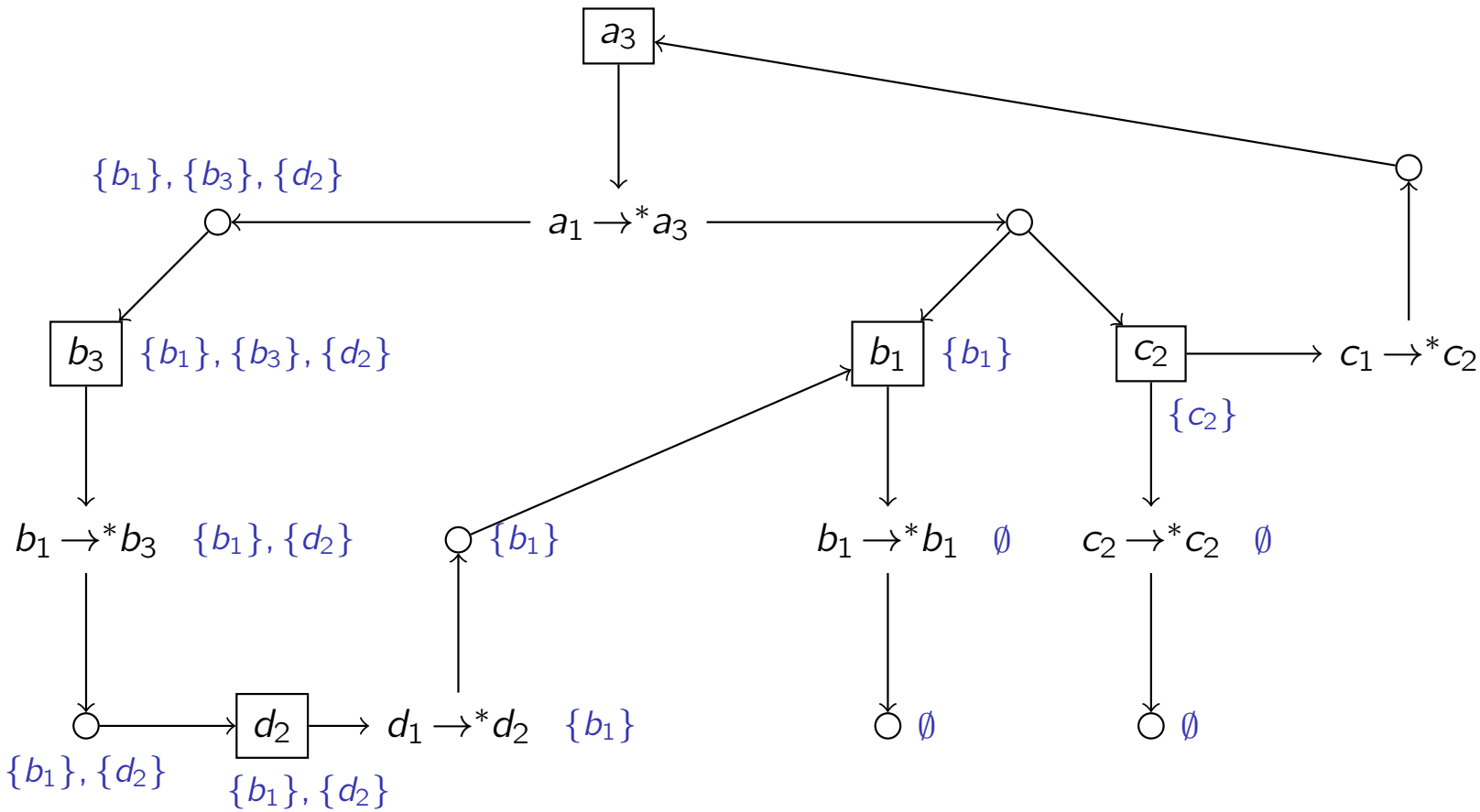


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

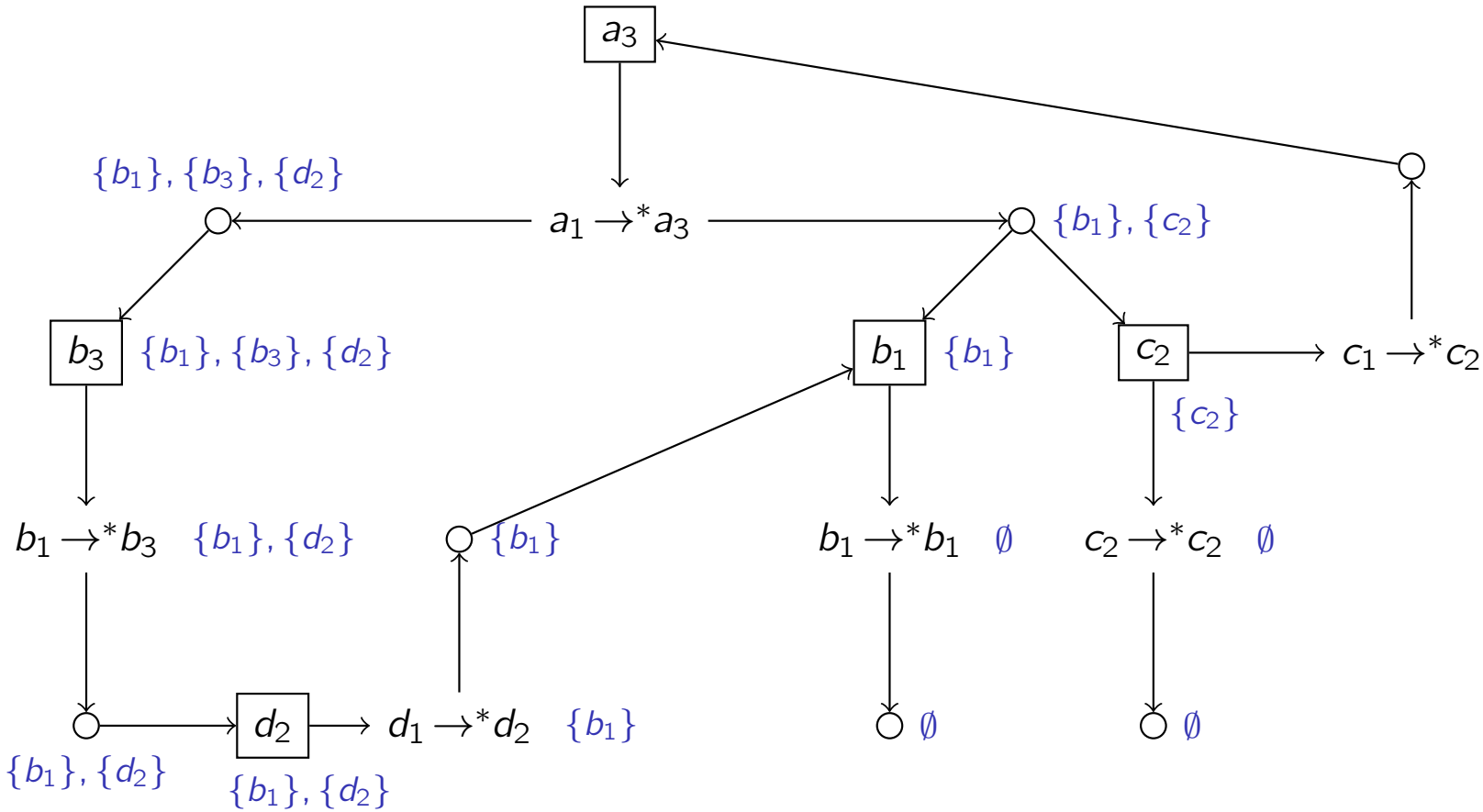


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

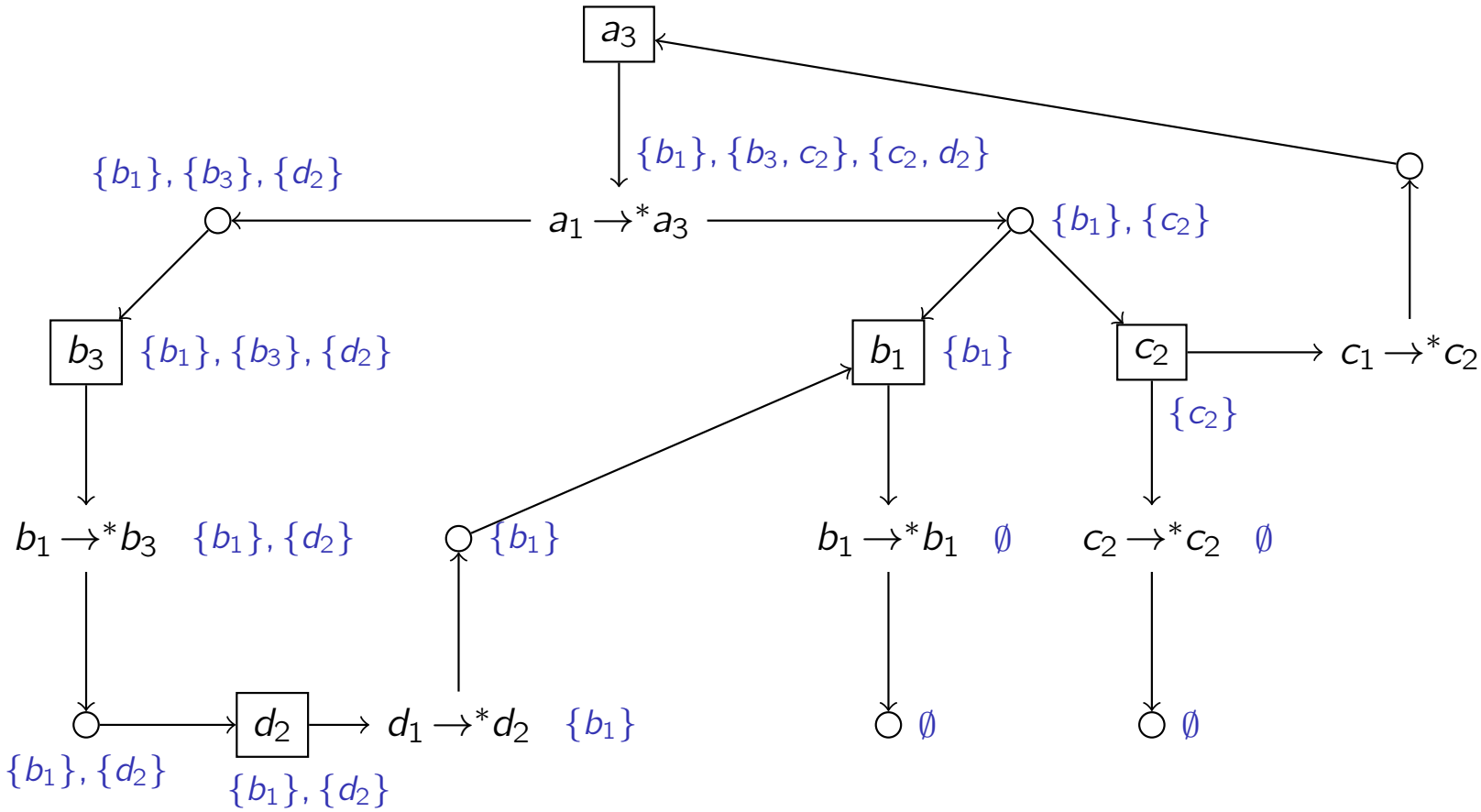


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

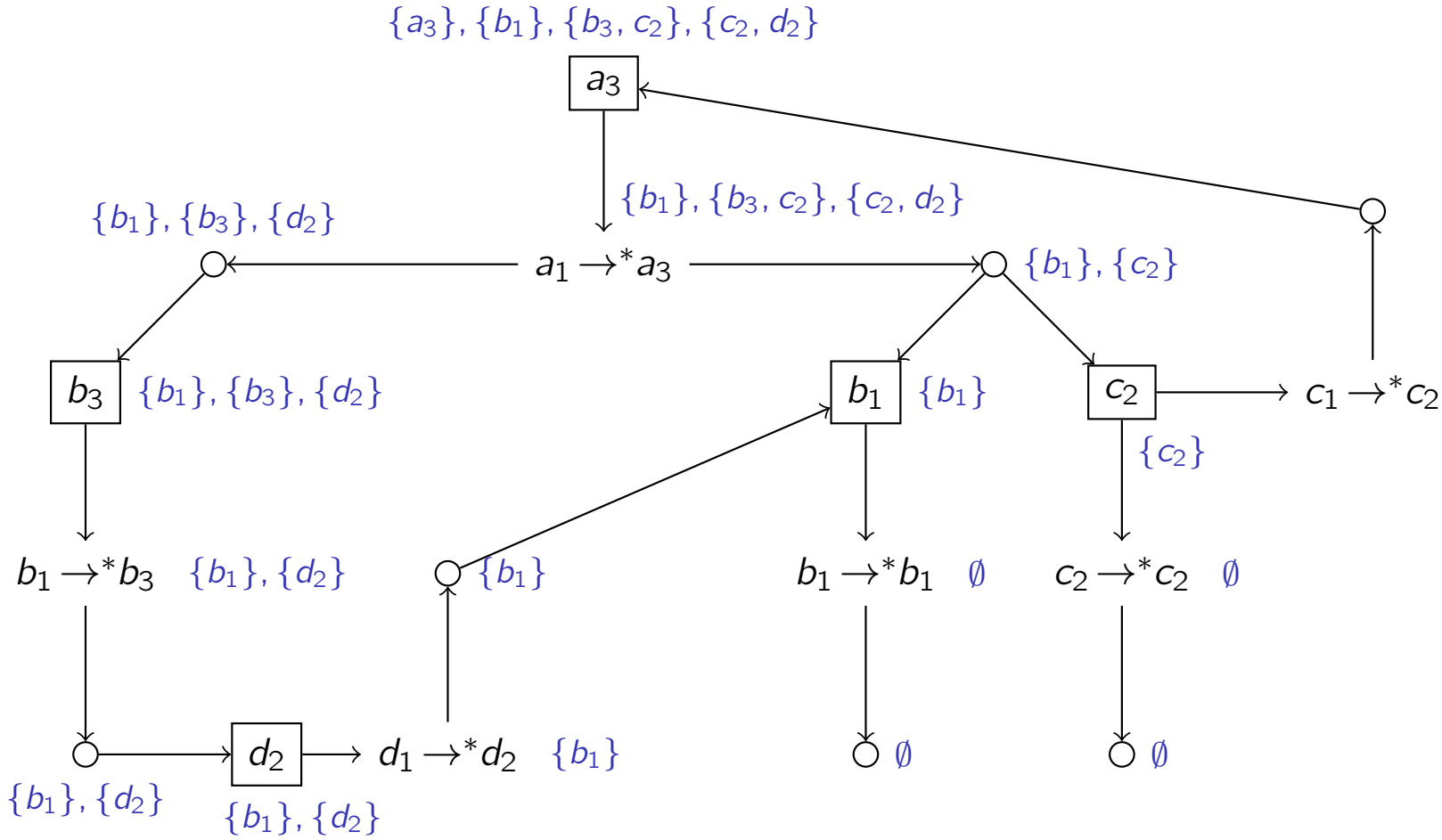


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

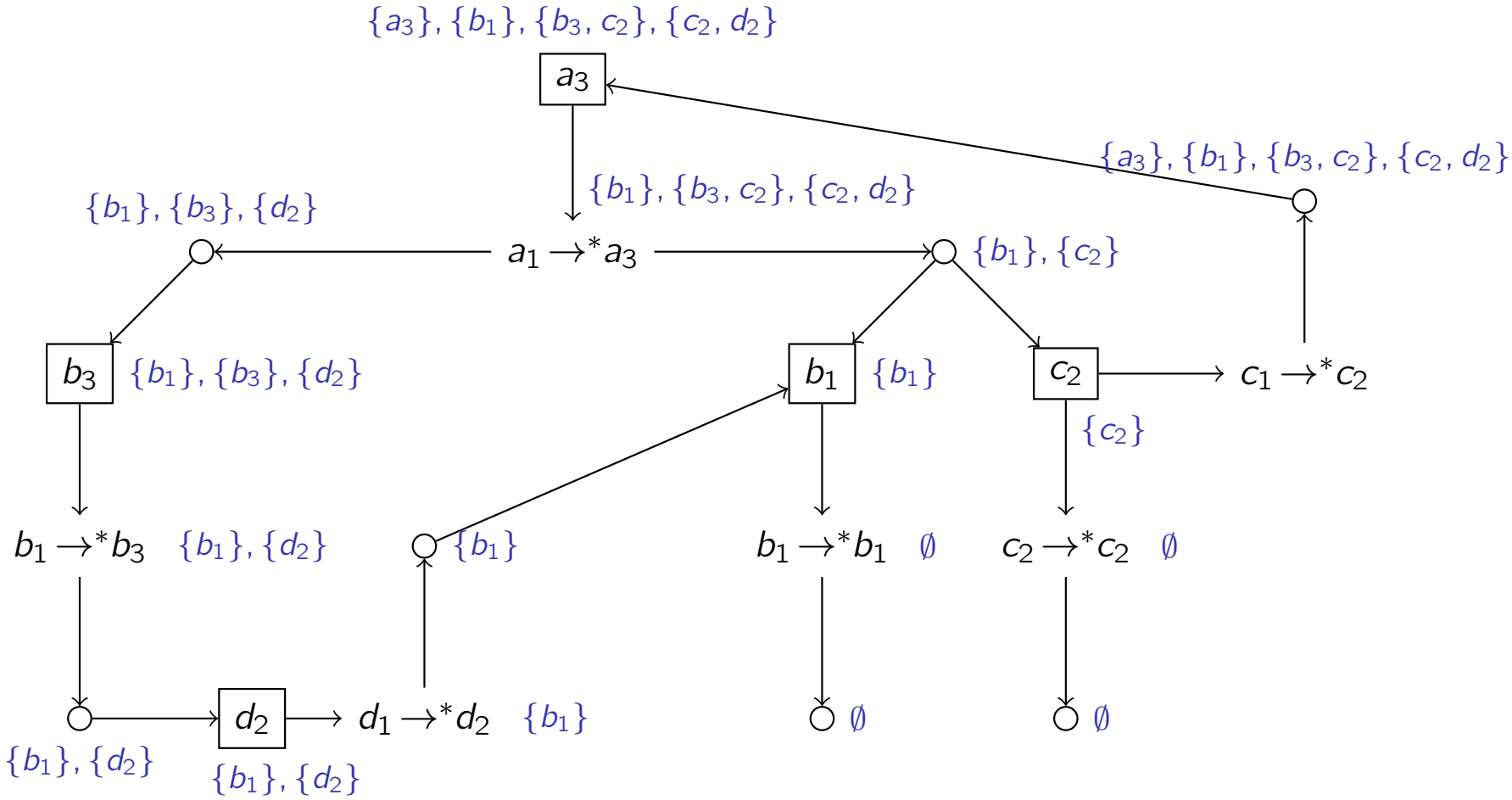


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

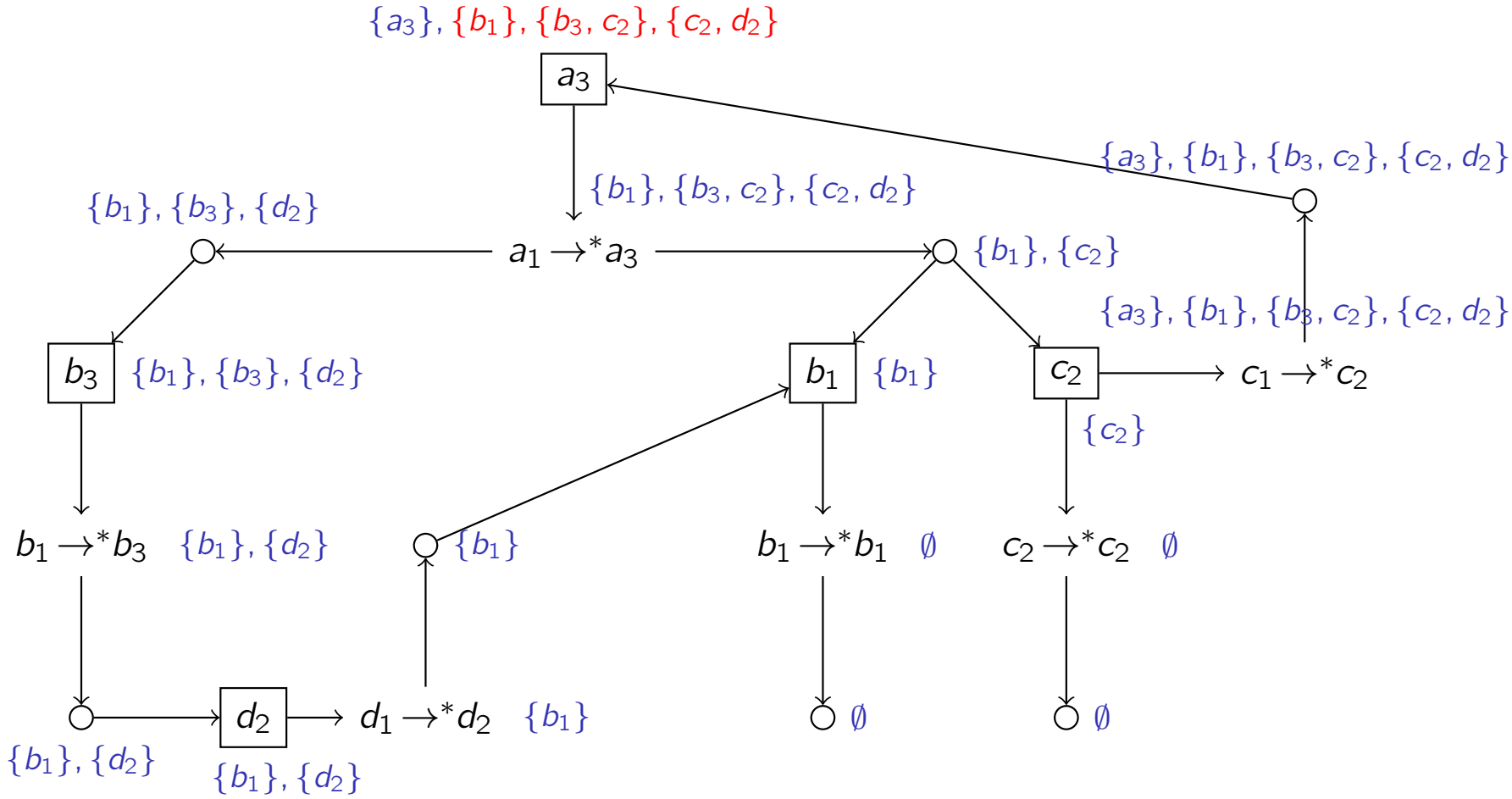


Cut Sets Under-approximation

Example

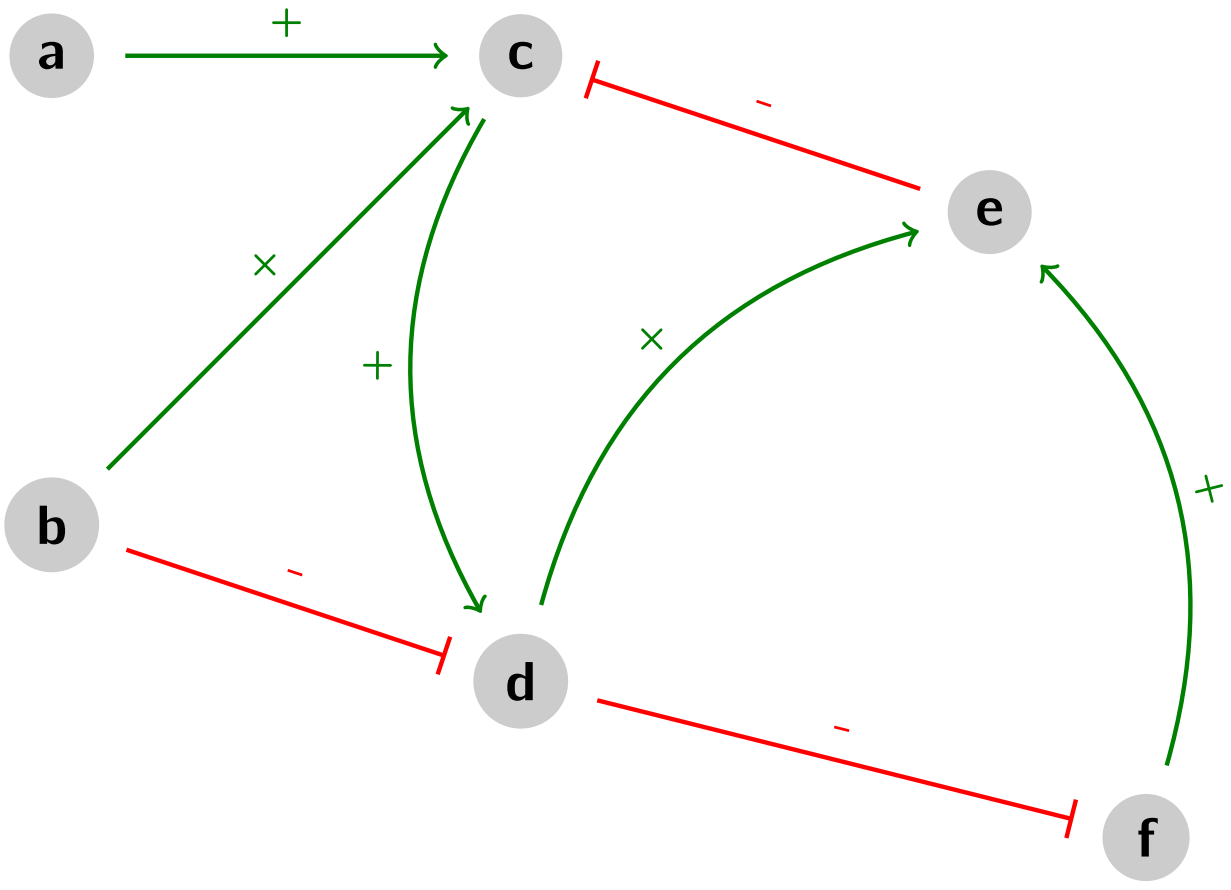
Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.



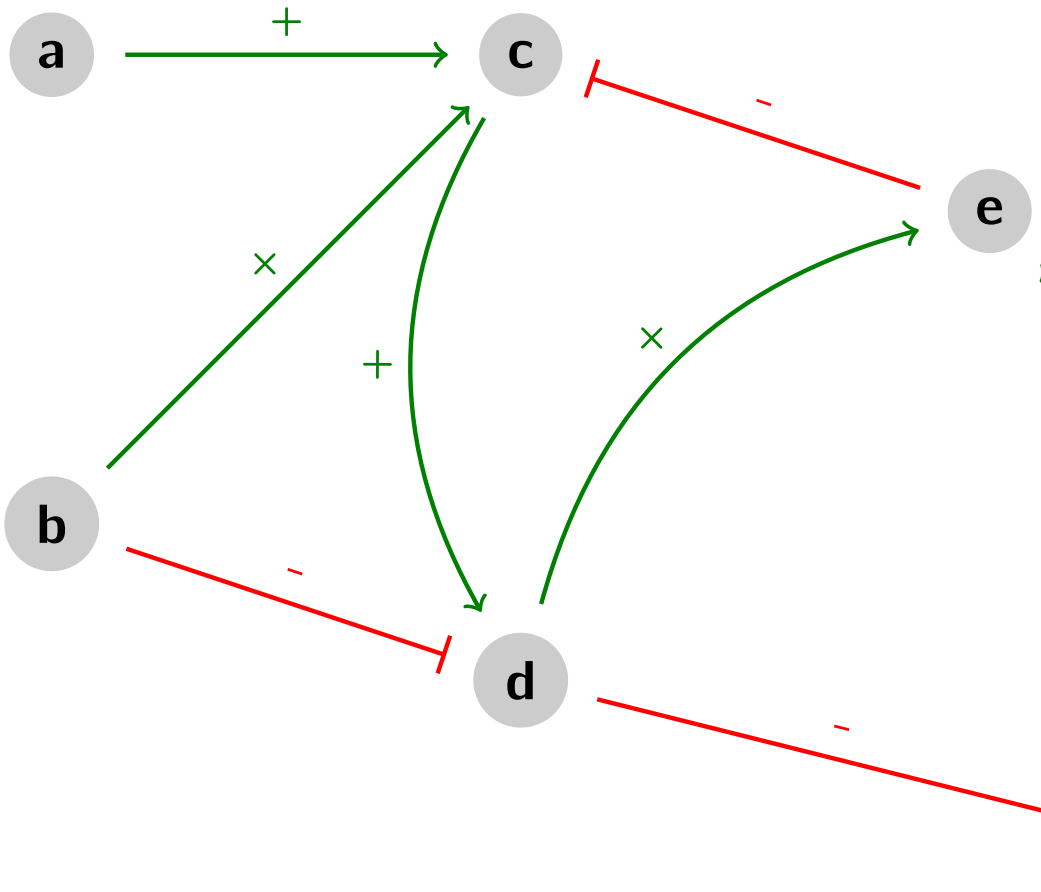
Biological Networks

E.g., Signalling Networks



Biological Networks

E.g., Signalling Networks



Components

Genes, proteins, complexes, ...

Activations

Positive influence (*a* increase may promote *c* increase).

Inhibitions

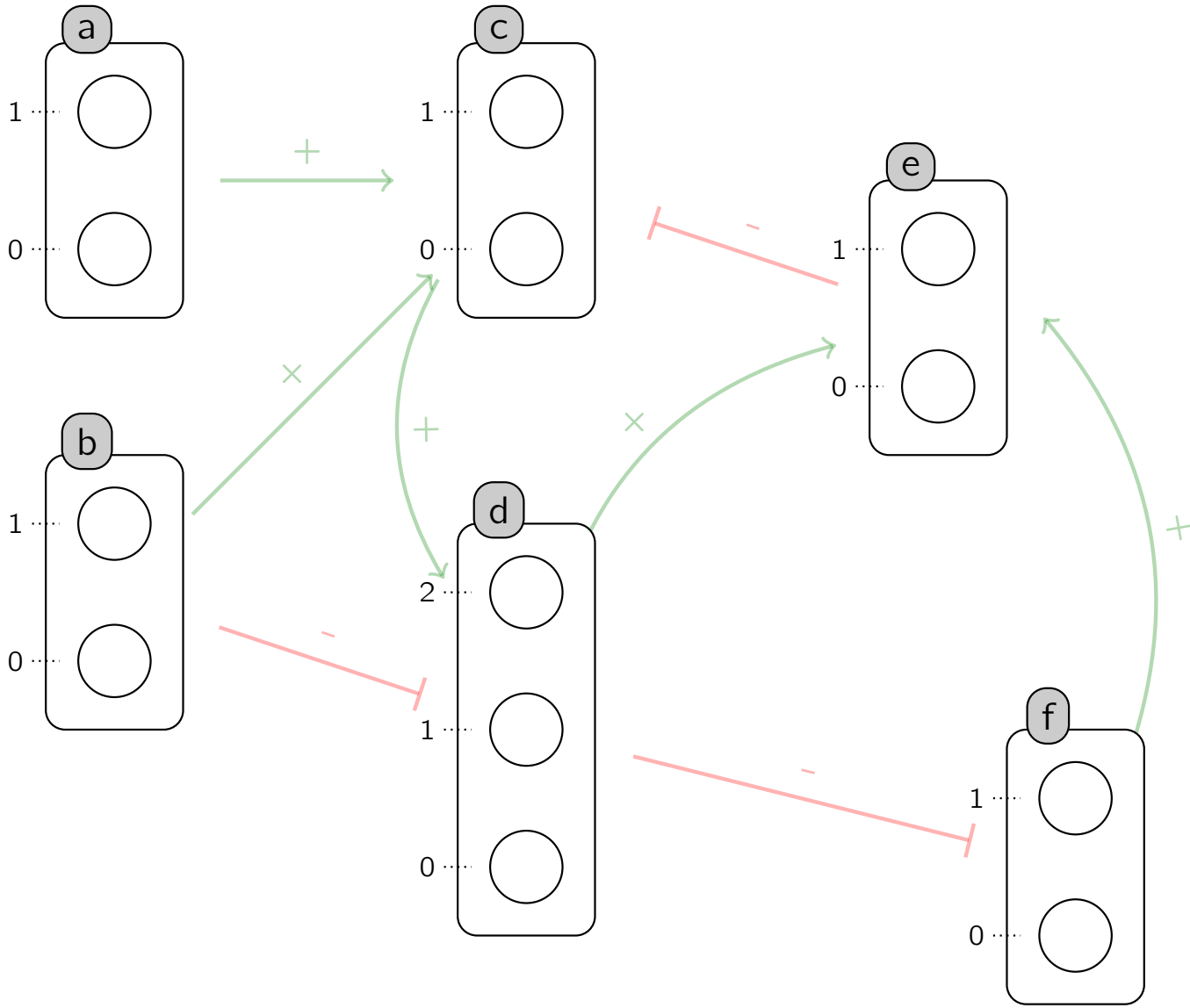
Negative influence (*b* increase may promote *d* decrease).

Typical settings

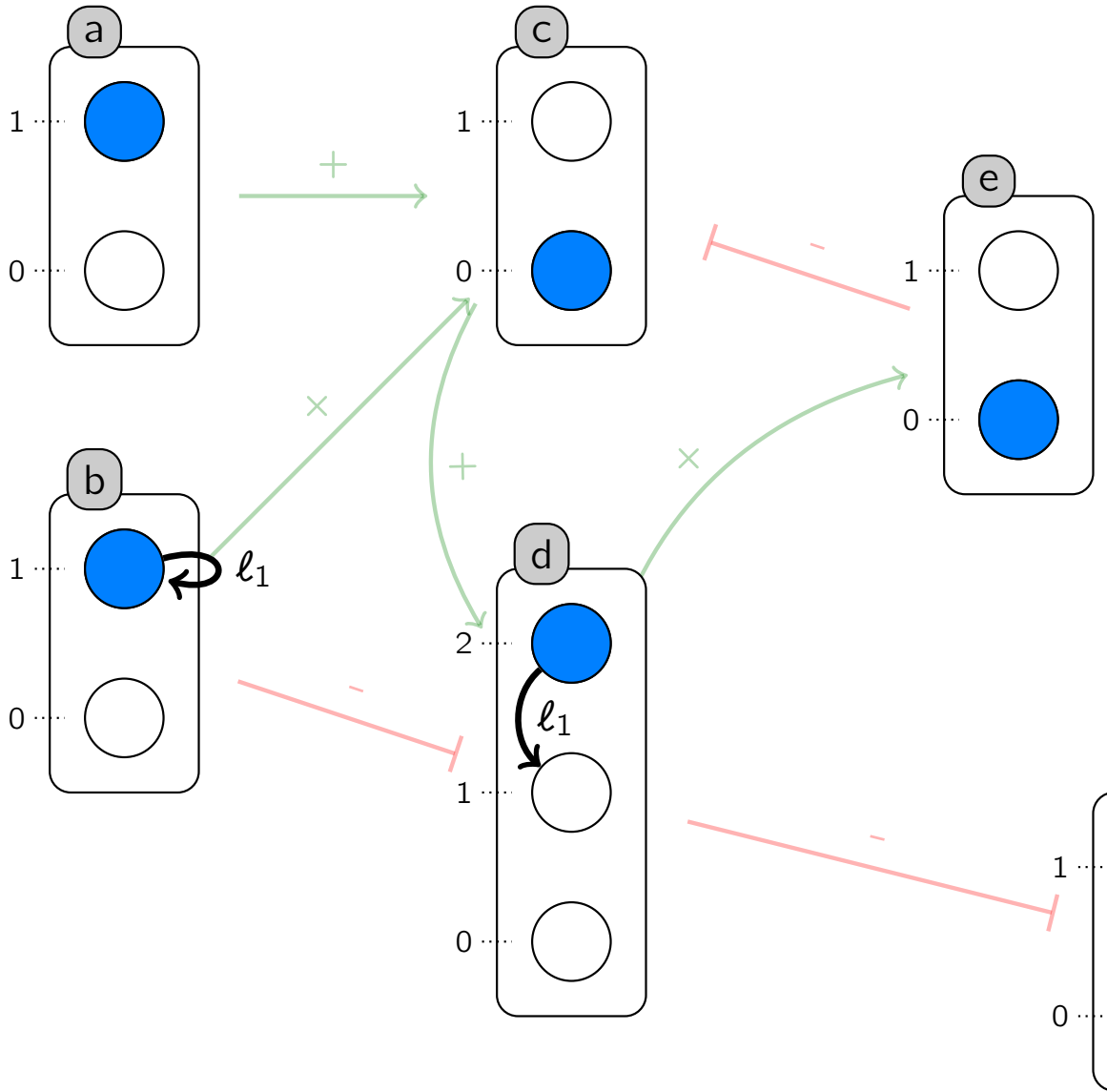
100 to +10,000 components

Few information on kinetics

Qualitative Models for Biological Networks



Qualitative Models for Biological Networks

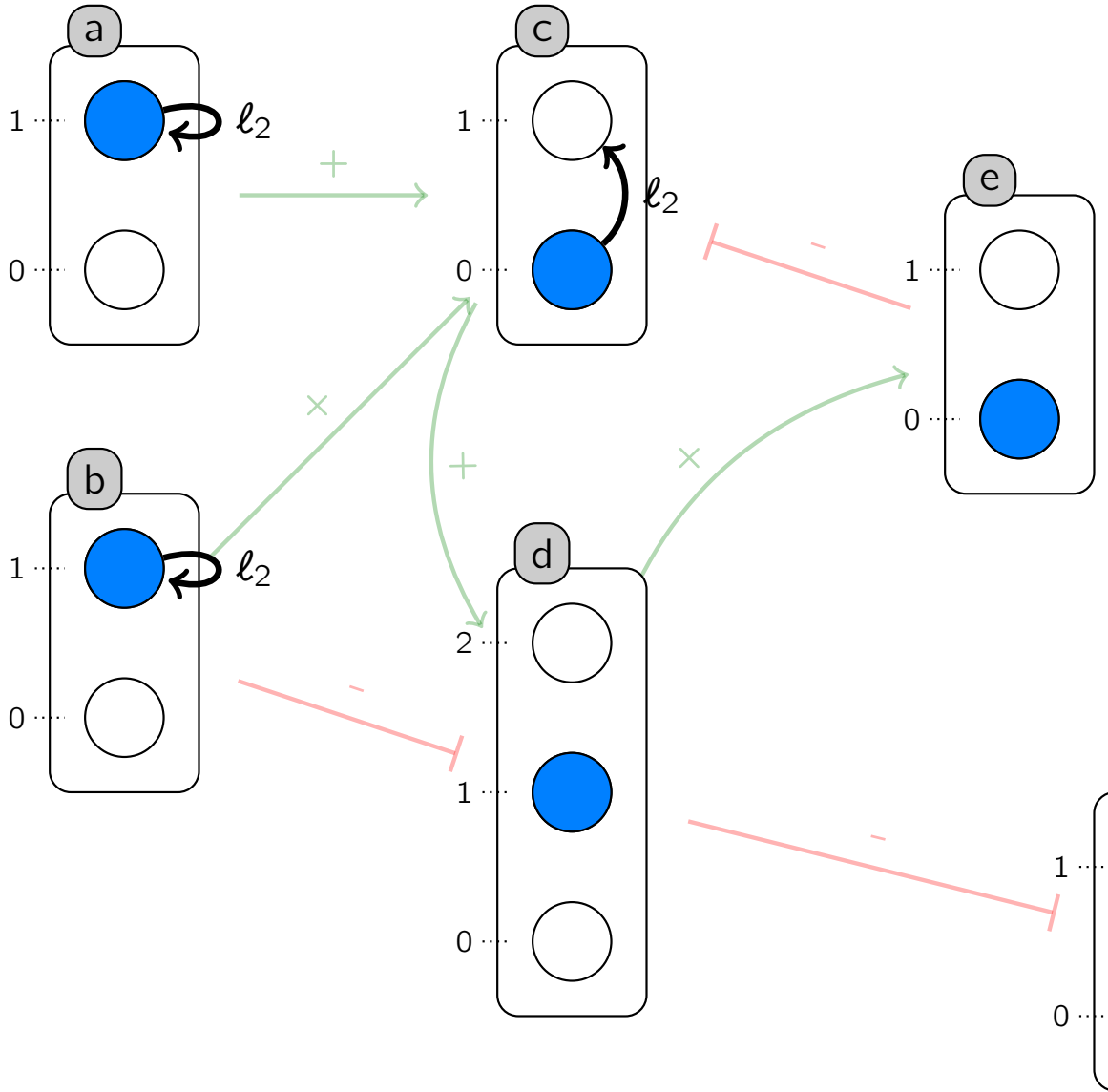


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.
Automata may be used to encode intermediate complexes.

Qualitative Models for Biological Networks

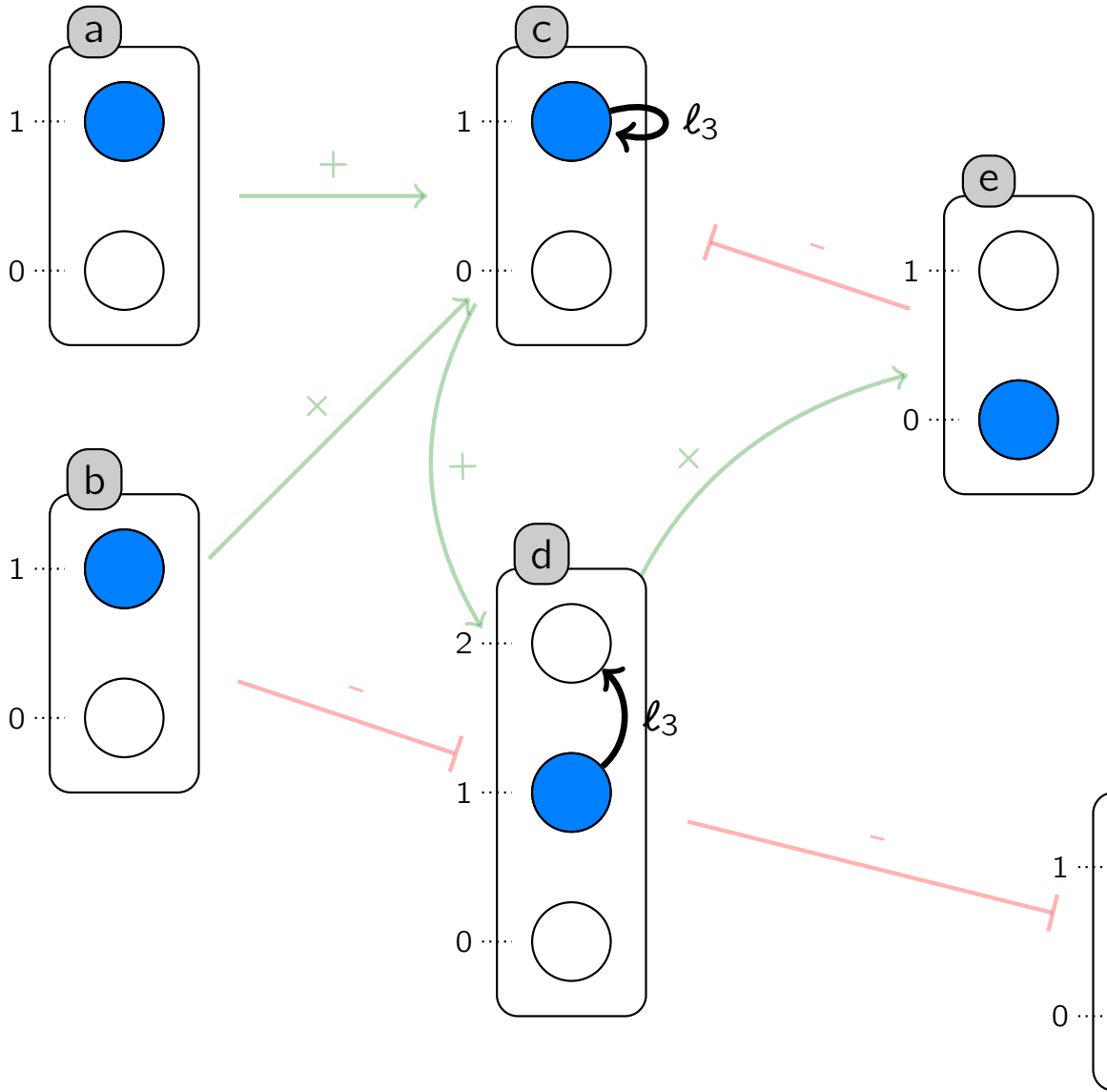


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.
Automata may be used to encode intermediate complexes.

Qualitative Models for Biological Networks

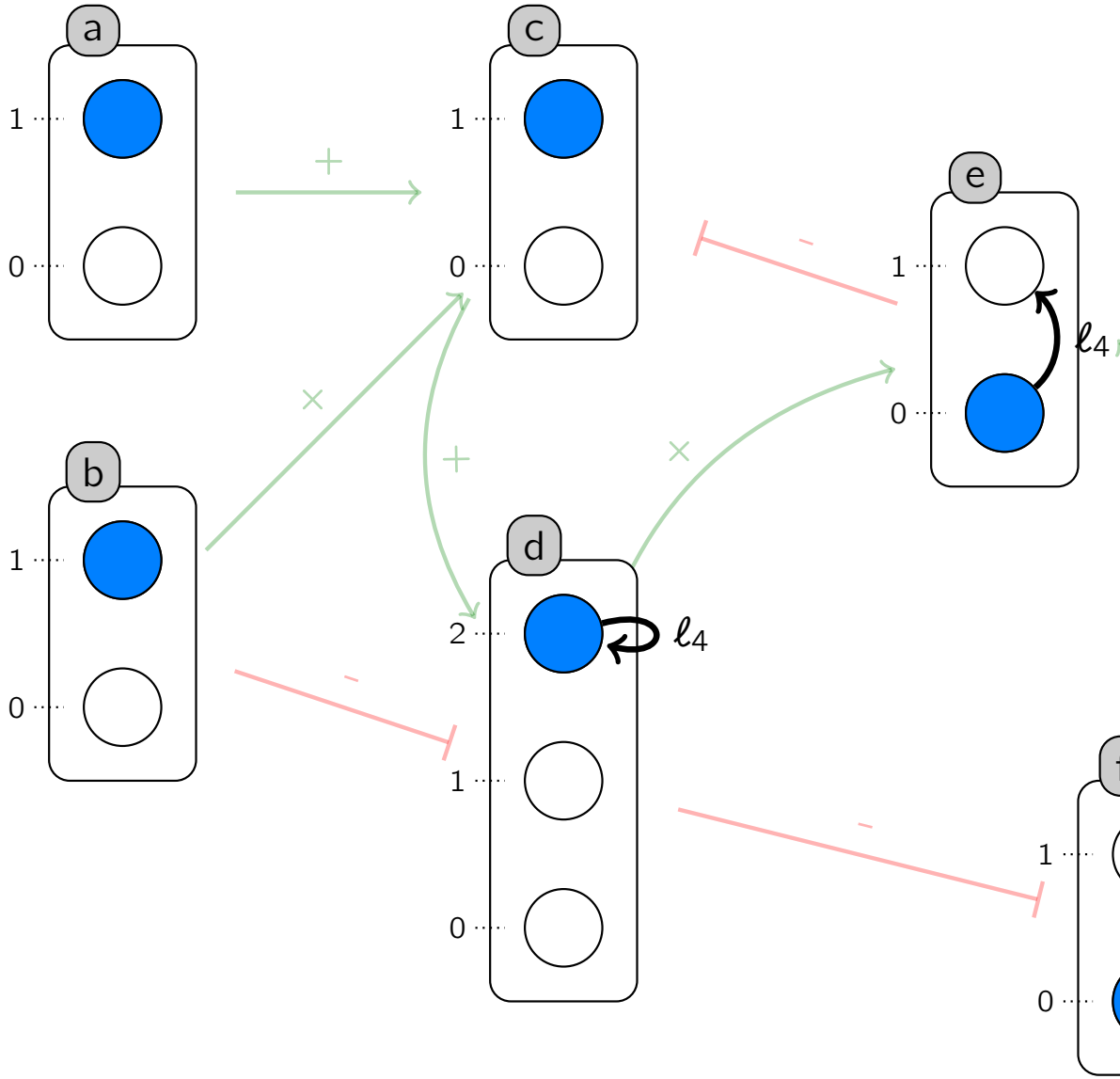


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.
Automata may be used to encode intermediate complexes.

Qualitative Models for Biological Networks

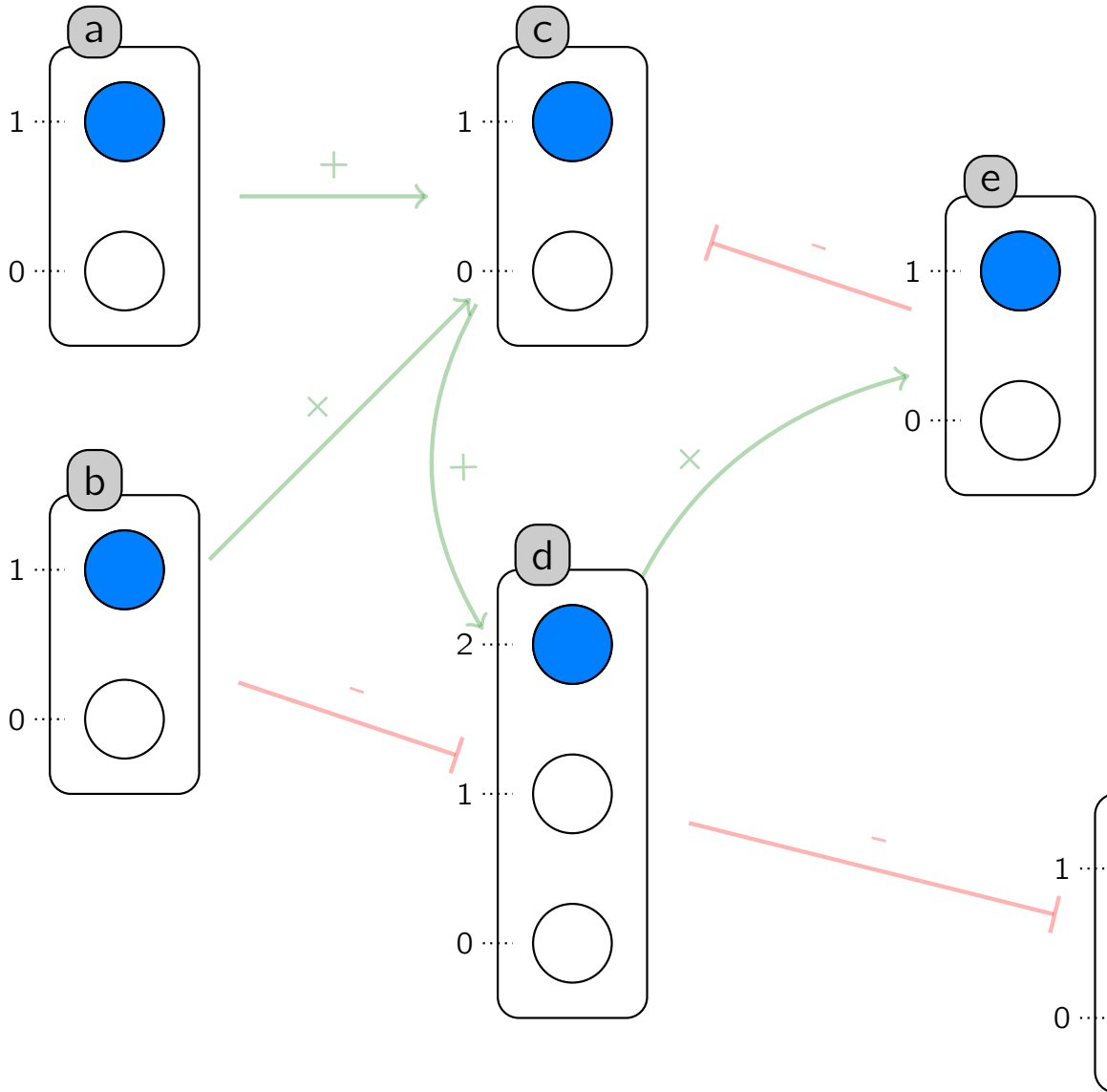


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.
Automata may be used to encode intermediate complexes.

Qualitative Models for Biological Networks

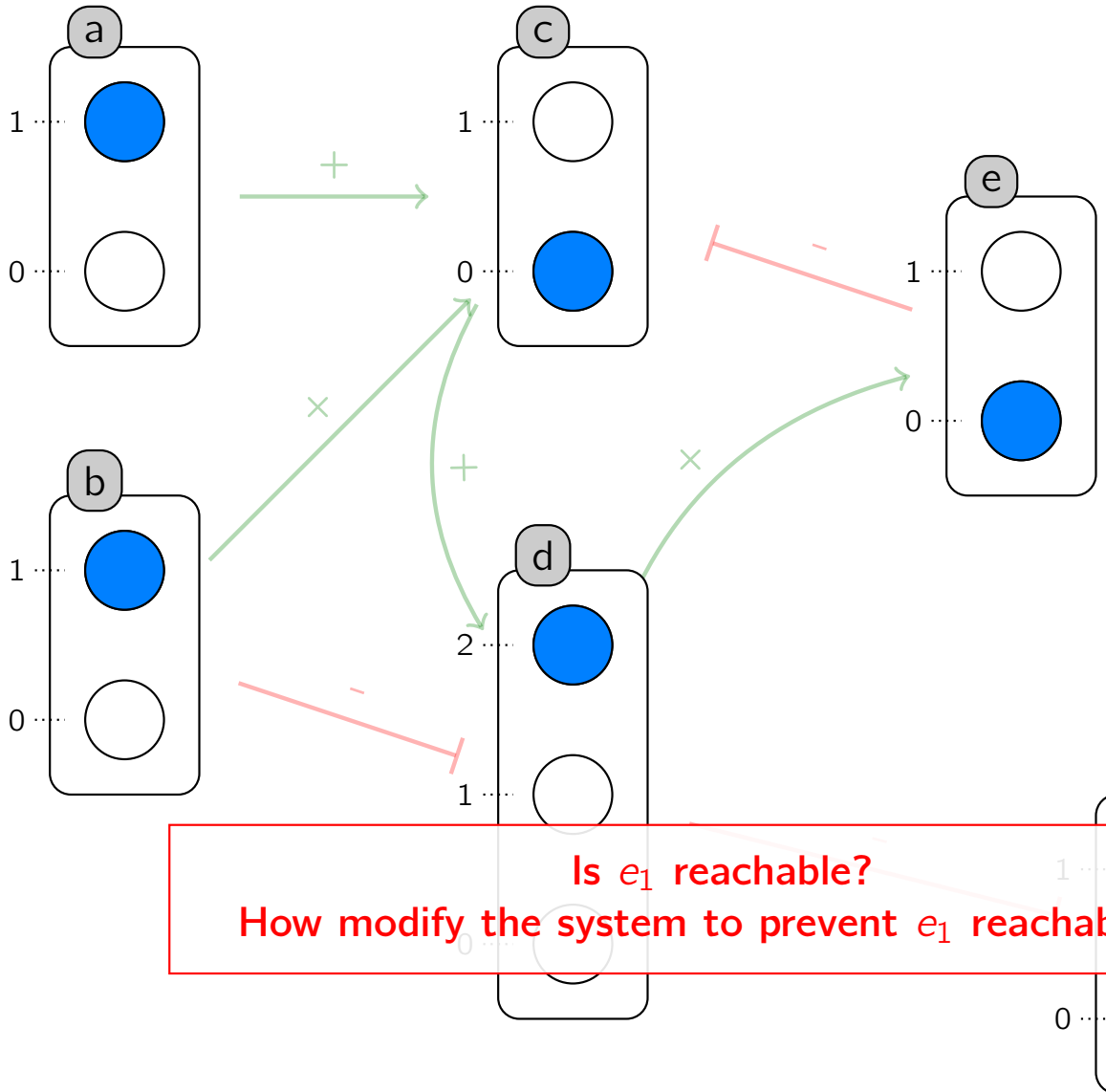


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.
Automata may be used to encode intermediate complexes.

Qualitative Models for Biological Networks



No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.
Automata may be used to encode intermediate complexes.

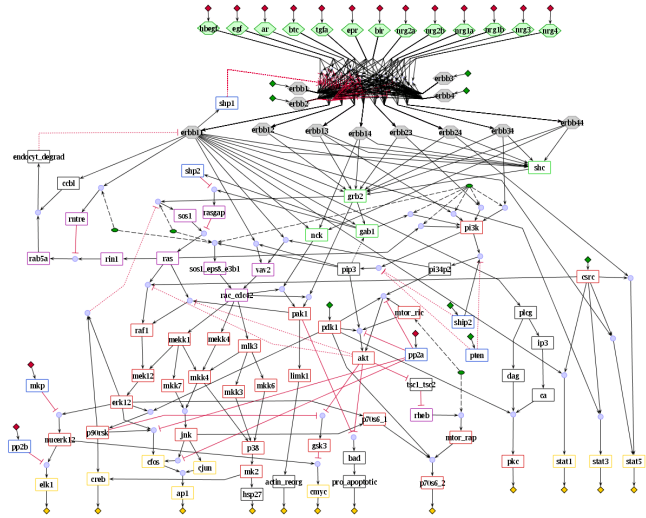
Is e_1 reachable?
How modify the system to prevent e_1 reachability?

Applications

Reachability analysis on large signalling networks

Model	NuSMV	libDDD	**PINT**
EGFR (35)	[3s-KO]	[1s-150s]	**0.007s**
TCR (54)	[1s-KO]	[0.6s-KO]	**0.004s**
TCR (133)	KO	KO	**0.030s**
EGFR (193)	KO	KO	**0.050s**

- wide-range of biological/arbitrary reachability
- always conclusive!

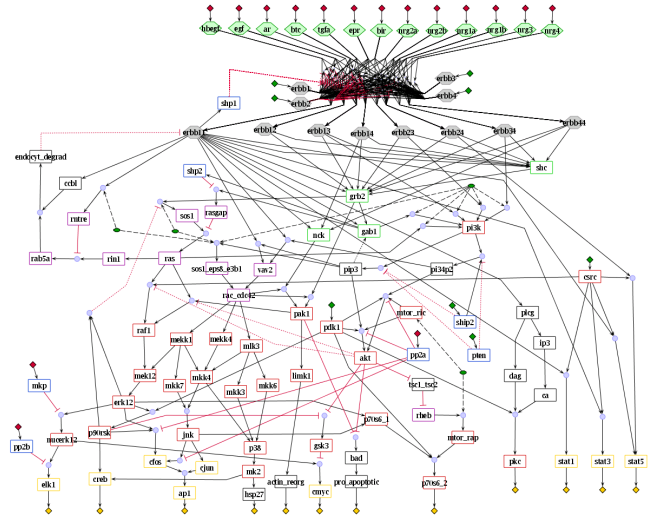


Applications

Reachability analysis on large signalling networks

Model	NuSMV	libDDD	**PINT**
EGFR (35)	[3s-KO]	[1s-150s]	**0.007s**
TCR (54)	[1s-KO]	[0.6s-KO]	**0.004s**
TCR (133)	KO	KO	**0.030s**
EGFR (193)	KO	KO	**0.050s**

- wide-range of biological/arbitrary reachability
- always conclusive!



Cut sets for very large networks extracted from PID

- 3 components of interest, very large context (environment).
- Async ANs with +21,000 automata.
- GLC: $\approx$ 20,000 nodes.
- Cut sets computation: up to a few minutes for cut sets w/ cardinality ≤ 5.
- No known alternative tractable on such large networks.

Discussion

Summary

- Graph of Local Causality (GLC): abstract representation of traces.
- Size of GLC is always reasonnable when few local states/transitions per automaton.
- Necessary/sufficient conditions on GLC for reachability.
- Cut sets for reachability from a set of initial states.

Applications

- Biological networks (qualitative models).
- Tractable on very large networks.
- Related work: transitions with priority classes (bio: time scales).

Future work

- Formal abstraction relationship with unfoldings / event structures.
- Identify attractors using the GLC.

Thank you for your attention.