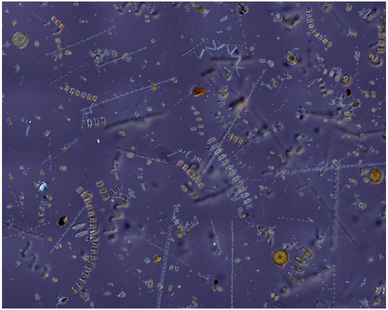
Interaction Graphs of Phytoplankton Species Interactions using Logical Learning

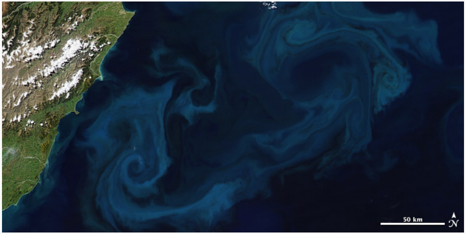
CMSB 2025

Madeleine Eyraud[1] Maxime Folschette[1] Katsumi Inoue[2]
Sébastien Lefebvre[3] Cédric Lhoussaine[1]

[1]Univ. Lille, CNRS, Centrale Lille, CRIStAL UMR 9189
[2]National Institute of Informatics, Tokyo
[3]Univ. Lille, CNRS, Univ. Littoral Côte d'Opale, LOG UMR 8187

Context: Phytoplankton and Ecosystem Dynamics





- Phytoplankton form the **base of marine trophic networks**
- Influence key processes e.g. **nutrient cycling** and **water quality**

⇒ **There is a strong interest in understanding phytoplankton dynamics**

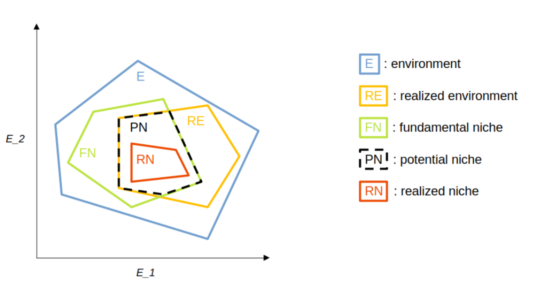
Context: Abiotic vs. Biotic Factors

- **Abiotic factors** (temperature, nutrients, ...) have **well-known effects** on phytoplankton growth

[1] Karasiewicz, S., Dolédec, S., Lefebvre, S.: Within outlying mean indexes: refining the omi analysis for the realized niche decomposition. PeerJ 5(e3364) (2017)

Context: Abiotic vs. Biotic Factors

- **Abiotic factors** (temperature, nutrients, ...) have **well-known effects** on phytoplankton growth
- **Biotic interactions** (between species) have only been **suggested**[1]



⇒We want to understand biotic interactions

[1] Karasiewicz, S., Dolédec, S., Lefebvre, S.: Within outlying mean indexes: refining the omi analysis for the realized niche decomposition. PeerJ 5(e3364) (2017)

- ODE
- Statistical models
- Machine learning

[5] Guziolowski, C., Videla, S., Eduati, F., Thiele, S., Cokelaer, T., Siegel, A., Saez-Rodriguez, J.: Exhaustively characterizing feasible logic models of a signaling network using answer set programming. Bioinformatics 30 (2013)

[6] Verny, L., Sella, N., Affeldt, S., Singh, P., Isambert, H.: Learning causal networks with latent variables from multivariate information in genomic data. PLOS Computational Biology 13, e1005662 (2017)

[7] Chevalier, S., Froidevaux, C., Paulevé, L., Zinovyev, A.: Synthesis of boolean networks from biological dynamical constraints using answer-set programming. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). pp. 34–41 (2019)

- ODE
- Statistical models
- Machine learning

⇒**We need explainable models**

[5] Guziolowski, C., Videla, S., Eduati, F., Thiele, S., Cokelaer, T., Siegel, A., Saez-Rodriguez, J.: Exhaustively characterizing feasible logic models of a signaling network using answer set programming. Bioinformatics 30 (2013)

[6] Verny, L., Sella, N., Affeldt, S., Singh, P., Isambert, H.: Learning causal networks with latent variables from multivariate information in genomic data. PLOS Computational Biology 13, e1005662 (2017)

[7] Chevalier, S., Froidevaux, C., Paulevé, L., Zinovyev, A.: Synthesis of boolean networks from biological dynamical constraints using answer-set programming. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). pp. 34–41 (2019)

Context: Modeling Approaches in Marine Ecosystems

- ODE
- Statistical models
- Machine learning

⇒We need explainable models

Symbolic Network Inference

- CASPO[5]
- MIIC[6]
- Bonesis[7]

---

[5] Guziolowski, C., Videla, S., Eduati, F., Thiele, S., Cokelaer, T., Siegel, A., Saez-Rodriguez, J.: Exhaustively characterizing feasible logic models of a signaling network using answer set programming. Bioinformatics 30 (2013)

[6] Verny, L., Sella, N., Affeldt, S., Singh, P., Isambert, H.: Learning causal networks with latent variables from multivariate information in genomic data. PLOS Computational Biology 13, e1005662 (2017)

[7] Chevalier, S., Froidevaux, C., Paulevé, L., Zinovyev, A.: Synthesis of boolean networks from biological dynamical constraints using answer-set programming. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). pp. 34–41 (2019)

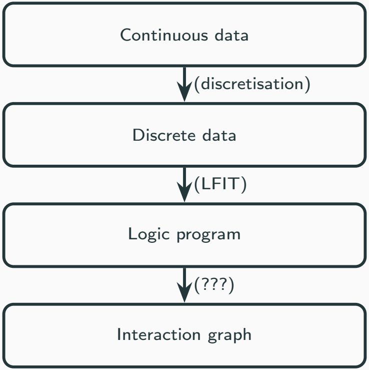
Problem & Contributions

Goal: Infer interpretable species interactions from long-term field data.

Key ideas:

- Species-specific, ecology-informed discretization
- Apply Learning From Interpretation Transition (LFIT)
- Mapping rules ⇒ signed, weighted interaction graph

Overview:



Dataset (SRN, Eastern English Channel)[2]

location	date	phaeocystis	ditylum	temperature	nitrate
001-P-015	1992-05-18	0.0	20.0	13	33
006-P-001	1992-06-12	6.0	100.0	14	30
...	...	...	...	...	...

- 1992–2020; sampling every 15–30 days
- **12 species**; **11 abiotic factors**
- **10 stations** on the french coast of the English Channel

[2] **SRN dataset - Regional Observation and Monitoring Program for Phytoplankton and Hydrology in the eastern English Channel (2025)**

Method: Species-specific discretization

Goal: translate continuous ecology into **discrete states** usable by LFIT—without losing species traits.

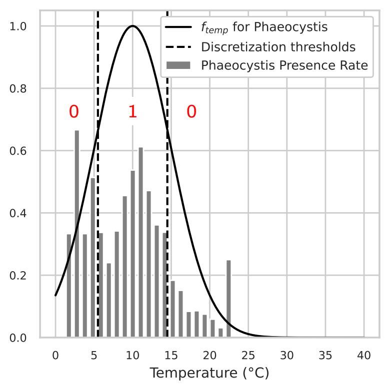
Method: Species-specific discretization

Goal: translate continuous ecology into **discrete states** usable by LFIT—without losing species traits.

Growth rate per species depending on temperature T:

$$f_{\text{temp}}(T) = \exp\left(-\frac{(T - T_{\text{opt}})^2}{2\sigma^2}\right)$$

- Favorable band: $T_{\text{opt}} \pm \sigma \Rightarrow$ state $= 1$
- Outside band $\Rightarrow$ state $= 0$



Temperature: theoretical response vs. observed presence.

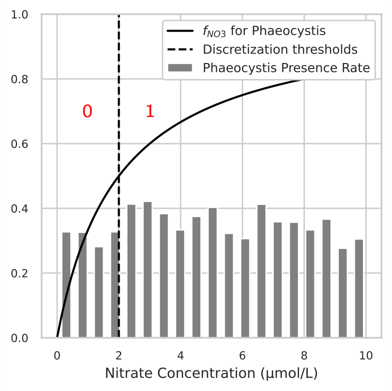
Method: Species-specific discretization

Goal: translate continuous ecology into **discrete states** usable by LFIT—without losing species traits.

Growth rate per species depending on nutrient X:

$$f_X([X]) = \frac{[X]}{[X] + K_X}$$

- Above ⇒ 1 (sufficient)
- Below ⇒ 0 (limiting)



Nitrate: theoretical response vs. observed presence.

⇒ **Species "see" the environment through their own physiological lenses.**

Method: Apply LFIT[3]

INPUT: set of state transitions

$$\begin{pmatrix} \text{phae} = 0 \\ \text{dit} = 1 \\ \text{temp} = 0 \\ \text{nit} = 0 \end{pmatrix} \rightarrow \begin{pmatrix} \text{phae} = 1 \\ \text{dit} = 1 \\ \text{temp} = 1 \\ \text{nit} = 0 \end{pmatrix} \qquad \begin{pmatrix} \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \vdots \end{pmatrix} \qquad \begin{pmatrix} \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \vdots \end{pmatrix} \qquad \begin{pmatrix} \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \vdots \end{pmatrix}$$

time: t $\rightarrow$ **t+1**

[3] Inoue, K., Ribeiro, T., Sakama, C.: Learning from interpretation transition. Machine Learning 94 (2014)

Method: Apply LFIT[3]

INPUT: set of state transitions

$$\begin{pmatrix} \text{phae} = 0 \\ \text{dit} = 1 \\ \text{temp} = 0 \\ \text{nit} = 0 \end{pmatrix} \rightarrow \begin{pmatrix} \text{phae} = 1 \\ \text{dit} = 1 \\ \text{temp} = 1 \\ \text{nit} = 0 \end{pmatrix} \qquad \begin{pmatrix} \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \vdots \end{pmatrix} \qquad \begin{pmatrix} \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \vdots \end{pmatrix} \qquad \begin{pmatrix} \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \vdots \end{pmatrix}$$

time: t → t+1

LFIT ↓

OUTPUT: logic program

$$\text{phae} = 1 \leftarrow \text{phae} = 0 \ \wedge \ \text{nit} = 0$$

$$\text{dit} = 1 \leftarrow \text{phae} = 0 \ \wedge \ \text{dit} = 1 \ \wedge \ \text{temp} = 0$$

$$\vdots$$

time: t+1 ← t

[3] Inoue, K., Ribeiro, T., Sakama, C.: Learning from interpretation transition. Machine Learning 94 (2014)

LFIT algorithms

- **GULA**[10]: complete but has exponential complexity — not scalable to our dataset.
- **PRIDE**[11]: polynomial-time but incomplete — prioritizes variables early in the input order.

[10] Ribeiro, T., Folschette, M., Magnin, M., Inoue, K.: Learning any memory-less discrete semantics for dynamical systems represented by logic programs. Machine Learning 111, 1–78 (2021)

[11] Ribeiro, T., Folschette, M., Magnin, M., Inoue, K.: Polynomial algorithm for learning from interpretation transition. 1st International Joint Conference on Learning & Reasoning (2021).

LFIT algorithms

- **GULA**[10]: complete but has exponential complexity — not scalable to our dataset.
- **PRIDE**[11]: polynomial-time but incomplete — prioritizes variables early in the input order.

Multiple runs & aggregation

- 5 runs with different variable orders (abiotic first to reduce spurious biotic links).
- Aggregate = union of minimal rules ⇒ improve coverage.
- **Aggregated model accuracy: 0.86**, higher than any single run (0.67–0.68).

[10] Ribeiro, T., Folschette, M., Magnin, M., Inoue, K.: Learning any memory-less discrete semantics for dynamical systems represented by logic programs. Machine Learning 111, 1–78 (2021)

[11] Ribeiro, T., Folschette, M., Magnin, M., Inoue, K.: Polynomial algorithm for learning from interpretation transition. 1st International Joint Conference on Learning & Reasoning (2021).

Method: From Rules to Influence Graph

Motivation:

- The logic program is explainable — but can contain **thousands of rules**.
- We propose to extract a **directed, weighted interaction graph** that summerize the rules.
- This graph provides a compact, readable view of the system's dynamics.

Method: From Rules to Influence Graph

Motivation:

- The logic program is explainable — but can contain **thousands of rules**.
- We propose to extract a **directed, weighted interaction graph** that summerize the rules.
- This graph provides a compact, readable view of the system's dynamics.

For each rule r:

- *coverage*(r) = # transitions with body true;
  support(r) = # transitions with body at t and head at $t+1$.
- Confidence $P(\text{head}|\text{body}) = \text{support}/\text{coverage}$.
- Rule weight: $w(r) = \text{support}(r) \cdot \frac{P(\text{head}|\text{body})}{P(\text{head})}$.

$$\underbrace{\text{phae} = 1}_{\text{head}} \leftarrow \underbrace{\text{phae} = 0 \ \wedge \ \text{nit} = 0}_{\text{body}}$$

Method: From Rules to Influence Graph

Method: From Rules to Influence Graph

$R_{i \to j}$ is the set of all rules where **Dit** appears in the **body** and **Phae** in the **head**:

$$\text{Phae}^1 \leftarrow \text{Dit}^1 \wedge \ldots$$

Method: From Rules to Influence Graph

$R_{i\to j}$ is the set of all rules where **Dit** appears in the **body** and **Phae** in the **head**:

$$\text{Phae}^1 \leftarrow \text{Dit}^1 \wedge \ldots$$

Edge **Thickness**:

$$\textstyle\sum_{r\in R_{i\to j}} w(r).$$

Method: From Rules to Influence Graph

$R_{i \to j}$ is the set of all rules where **Dit** appears in the **body** and **Phae** in the **head**:

$$\text{Phae}^1 \leftarrow \text{Dit}^1 \wedge \ldots$$

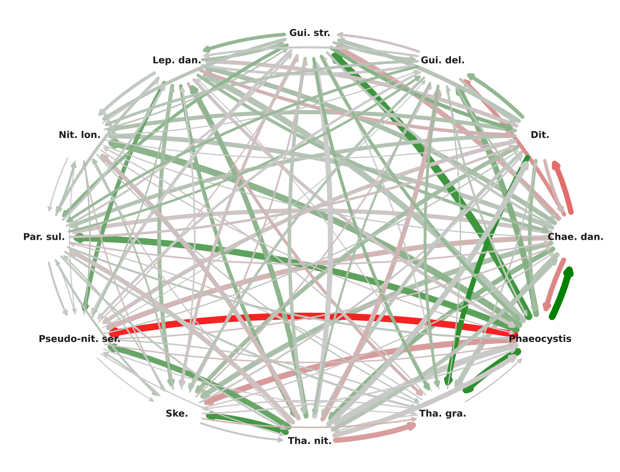
Edge **Thickness**:

$$\textstyle\sum_{r \in R_{i \to j}} w(r).$$

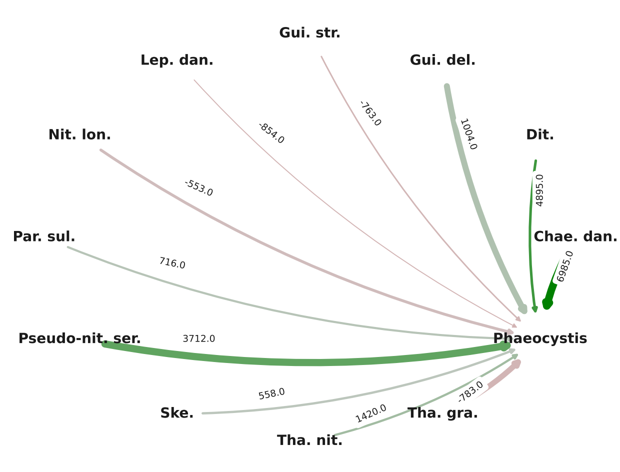
Edge Color:

$\delta_{i \to j}(r) \in \{\pm 1\}$ (alignment vs contrast)
and $\sum_r \delta_{i \to j}(r)\, w(r)$ maps to color.

Results: Phytoplankton Influence Graph

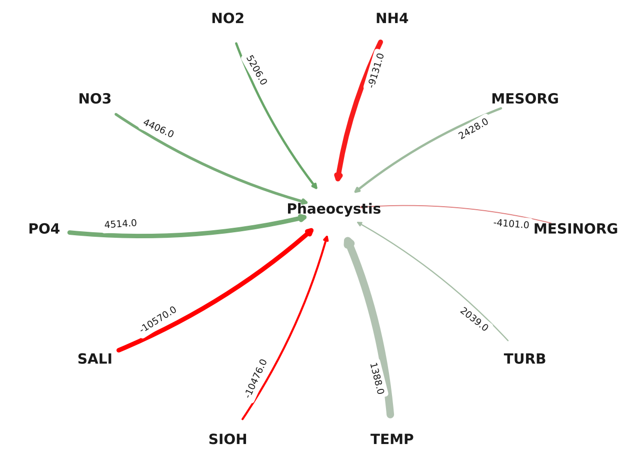


Results: *Phaeocystis*



Edges normalized locally

Results: Abiotic influences on *Phaeocystis*



NO_2 & PO_4 positive; $Si(OH)_4$ negative; salinity & NH_4 negative — consistent with bloom phenology.

Discussion

Contributions

- Pipeline for applying LFIT on ecological time series
- Influence graphs summarizing thousands of rules into readable interactions

Limits

- Graphs show *influence* patterns; they're not direct interaction types (competition, allelopathy, etc.)
- Memoryless learning; not causal; requires ecological interpretation

Resources

- Code (notebooks): `https://zenodo.org/records/15389109`

Ongoing work: Combining abiotic factors into a single response

- Build a *theoretical growth response* R_s for each species s by combining eco-physiological functions of abiotic drivers, then use it as a single feature
- $R_s = f_{\mathrm{T}}^{(s)} \cdot \min\left(f_{\mathrm{lum}}^{(s)}, f_{n_1}^{(s)}, f_{n_2}^{(s)}, \ldots\right)$

$\Rightarrow$ Replace many abiotics with a single, species-specific theoritical response capturing the eco-theory.

$\Rightarrow$ Reduce the number of variables, potentially the number of rules.

Ongoing work: Rules analysis

Restrict to rules with same head atom: phae=1 $\leftarrow$ phae= $0 \wedge$ nit= 0

Ongoing work: Rules analysis

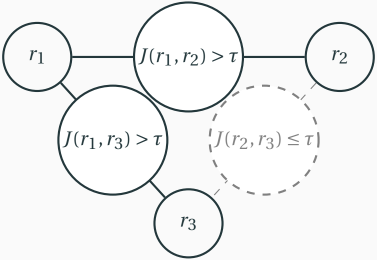
Restrict to rules with same head atom: phae=1 $\leftarrow$ phae$=0 \wedge$ nit$=0$

Calculate similarity index (Jaccard) between rules: $J(r_i, r_j) = \frac{|\text{body}(r_i) \cap \text{body}(r_j)|}{|\text{body}(r_i) \cup \text{body}(r_j)|}$

Ongoing work: Rules analysis

Restrict to rules with same head atom: phae=1 $\leftarrow$ phae= 0 $\wedge$ nit= 0

Calculate similarity index (Jaccard) between rules: $J(r_i, r_j) = \frac{|\text{body}(r_i) \cap \text{body}(r_j)|}{|\text{body}(r_i) \cup \text{body}(r_j)|}$

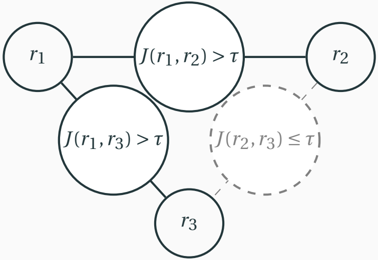


$G = (V, E)$ with an edge $(i, j) \in E$ iff $J(r_i, r_j) > \tau$

Ongoing work: Rules analysis

Restrict to rules with same head atom: phae=1 $\leftarrow$ phae$=0 \wedge$ nit$=0$

Calculate similarity index (Jaccard) between rules: $J(r_i, r_j) = \frac{|\text{body}(r_i) \cap \text{body}(r_j)|}{|\text{body}(r_i) \cup \text{body}(r_j)|}$



$G = (V, E)$ with an edge $(i, j) \in E$ iff $J(r_i, r_j) > \tau$

- Detect communities in G
- In each community, extract the most common body patterns

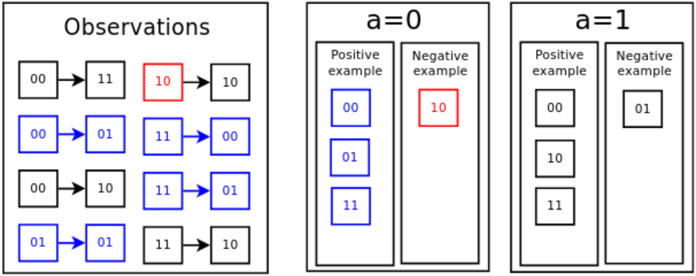
$\Rightarrow$ **Clearer "typical contexts" per head species, and stable summaries**

Thank you!

Learning From Interpretation Transition (LFIT)

Learning Algorithm Intuition: Classification Problem

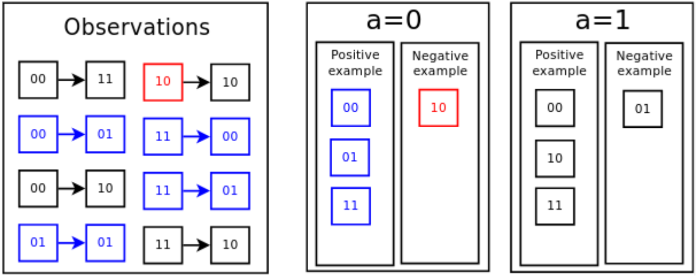
Learn applicable rules: conditions so that a variable **can** take a certain value in next state.



Equivalent to a **classification problem**: What is a typical state where a can take value 0 in the next state ? Here: when a_0 or b_1 is present.

Learning Algorithm Intuition: Classification Problem

Learn applicable rules: conditions so that a variable **can** take a certain value in next state.



Equivalent to a **classification problem**: What is a typical state where a can take value 0 in the next state ? Here: when a_0 or b_1 is present.

$$a_0 \leftarrow a_0. \qquad a_0 \leftarrow b_1.$$

Presentation of GULA

GULA = General Usage LFIT Algorithm

Input: a set of transitions $(s_1 \rightarrow s_2)$

Output: a logic program that respects:

- **Consistency**: the program allows no negative examples
- **Realization**: the program covers all positive examples
- **Completeness**: the program covers all the state space
- **Minimality** of the rules (most general conditions)

Method: start from most general rules and **specialize** iteratively.

Minimal refinements

Suppose: $\text{dom}(a) = \text{dom}(b) = \{0, 1\}$ and $\text{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

Minimal refinements

Suppose: $\text{dom}(a) = \text{dom}(b) = \{0, 1\}$ and $\text{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$$a_1 \leftarrow a_0, c_2.$$
$$a_1 \leftarrow b_1, c_2.$$
$$a_1 \leftarrow c_2, c_0.$$
$$a_1 \leftarrow c_2, c_1.$$

$$a_1 \leftarrow b_1.$$
(No change)

Minimal refinements

Suppose: $\text{dom}(a) = \text{dom}(b) = \{0, 1\}$ and $\text{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$$a_1 \leftarrow a_0, c_2.$$
$$a_1 \leftarrow b_1, c_2.$$
$$a_1 \leftarrow c_2, c_0.$$
$$a_1 \leftarrow c_2, c_1.$$

$$a_1 \leftarrow b_1.$$

Minimal refinements

Suppose: $\mathrm{dom}(a) = \mathrm{dom}(b) = \{0, 1\}$ and $\mathrm{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$a_1 \leftarrow c_2.$ $\qquad$ $a_1 \leftarrow b_1.$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$a_1 \leftarrow a_0, c_2.$

$a_1 \leftarrow b_1, c_2.$

$a_1 \leftarrow b_1.$
(More general)

Minimal refinements

Suppose: $\text{dom}(a) = \text{dom}(b) = \{0, 1\}$ and $\text{dom}(c) = \{0, 1, 2\}$ and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$$a_1 \leftarrow a_0, c_2. \qquad\qquad a_1 \leftarrow b_1.$$

Results

Tony Ribeiro, Maxime Folschette, Morgan Magnin and Katsumi Inoue. **Learning any memory-less discrete semantics for dynamical systems represented by logic programs**. *Machine Learning* 111, Springer. November 2021. `https://doi.org/10.1007/s10994-021-06105-4`

- **Allows to learn the network** (structure of the model)
- **Independent of the semantics**
  (characterization of applicable memoryless semantics)

Nice in theory, but in practice?

- **Exponential complexity** → How to handle big datasets?
  (many transitions, many variables)
- **Exact learning** → How to handle noise?

Two Heuristic on LFIT

Weighted Likeliness/Unlikeliness Rules

- Use the algorithm twice to learn two logic programs:
  - likeliness rules: what is possible
  - unlikeliness rules: what is impossible
- Weight each rule by the number of observations it matches

Statistical overlay $\Rightarrow$ usable on **noisy datasets**

Likeliness rules

$(3, a_0 \leftarrow b_1)$

$(15, a_1 \leftarrow b_0)$

$\vdots$

Unlikeliness rules

$(30, a_0 \leftarrow c_1)$

$(5, a_1 \leftarrow c_0)$

$\vdots$

Using Weighted Likeliness/Unlikeliness Rules

Explainable predictions:

- Compare weights of applicable likeliness/unlikeliness rules
- Ratio of highest weights $\Rightarrow$ **probability** P
- Rules with highest weights $\Rightarrow$ **explanation** E

$$\text{predict} : (atom, state) \mapsto (P, E)$$

Likeliness rules

$(3, a_0 \leftarrow b_1)$

$(15, a_1 \leftarrow b_0)$

Unlikeliness rules

$(30, a_0 \leftarrow c_1)$

$(5, a_1 \leftarrow c_0)$

Using Weighted Likeliness/Unlikeliness Rules

Explainable predictions:

- Compare weights of applicable likeliness/unlikeliness rules
- Ratio of highest weights $\Rightarrow$ **probability** P
- Rules with highest weights $\Rightarrow$ **explanation** E

$$\text{predict} : (atom, state) \mapsto (P, E)$$

Likeliness rules	**Unlikeliness rules**
$(3, a_0 \leftarrow b_1)$	$(30, a_0 \leftarrow c_1)$
$(15, a_1 \leftarrow b_0)$	$(5, a_1 \leftarrow c_0)$

$$\text{predict}(a_1, \langle a_1, b_1, c_0 \rangle) = (0.75, ((15, a_1 \leftarrow b_0), (5, a_1 \leftarrow c_0))) \Rightarrow \text{Likely}$$

Using Weighted Likeliness/Unlikeliness Rules

Explainable predictions:

- Compare weights of applicable likeliness/unlikeliness rules
- Ratio of highest weights $\Rightarrow$ **probability** P
- Rules with highest weights $\Rightarrow$ **explanation** E

$$\text{predict} : (atom, state) \mapsto (P, E)$$

Likeliness rules	**Unlikeliness rules**
$(3, a_0 \leftarrow b_1)$	$(30, a_0 \leftarrow c_1)$
$(15, a_1 \leftarrow b_0)$	$(5, a_1 \leftarrow c_0)$

$\text{predict}(a_1, \langle a_1, b_1, c_0 \rangle) = (0.75, ((15, a_1 \leftarrow b_0), (5, a_1 \leftarrow c_0))) \Rightarrow$ Likely

$\text{predict}(a_0, \langle a_1, b_1, c_0 \rangle) = (0.09, ((3, a_0 \leftarrow b_1), (30, a_0 \leftarrow c_1))) \Rightarrow$ Unlikely