Learning any memory-less discrete semantics for dynamical systems represented by logic programs

Tony Ribeiro[1,3], Maxime Folschette[2], Morgan Magnin[1,3], Katsumi Inoue[3]

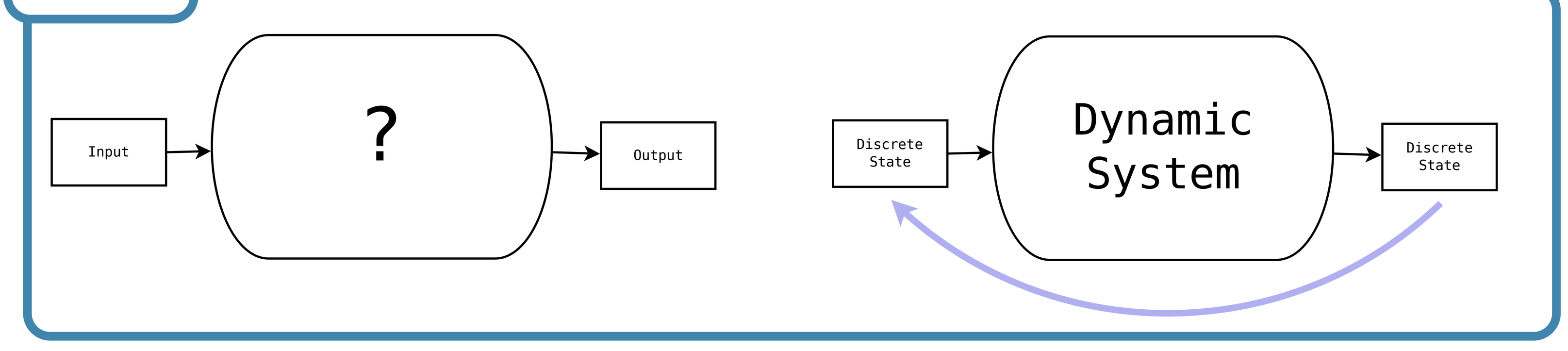
(1) Université de Nantes, Centrale Nantes, CNRS, LS2N, F-44000 Nantes, France (2) Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

(3) National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan tony.ribeiro@ls2n.fr, maxime.folschette@irisa.fr, morgan.magnin@ls2n.fr, inoue@nii.ac.jp

Motivations: Learning Dynamics

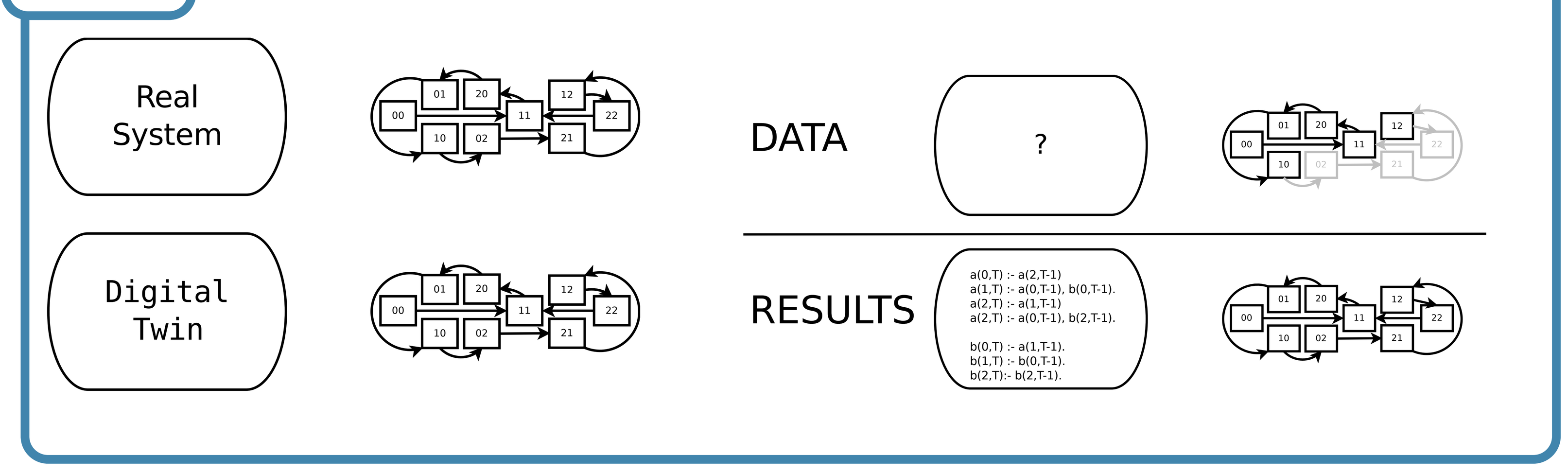
- Given a set of input/output states of a black-box system, learn its internal mechanics.
- **Discrete system:** input/output are vectors of same size which contain discrete values.
- **Dynamic system:** input/output are states of the system and output is the next input.

Problem



- **Goal:** produce an artificial system with the same behavior, i.e., a digital twin.
- **Representation:** propositional logic programs encoding multi-valued discrete variables.
- **Method:** learn the dynamics of systems from its state transitions.

Goal



Formalization: $\mathcal{MVL}$ and $\mathcal{DMVLP}$

Definition 1 (Atoms). *Let $\mathcal{V} = \{v_1, \dots, v_n\}$ be a finite set of $n \in \mathbb{N}$ variables, and $\text{dom} : \mathcal{V} \to \mathbb{N}$. The atoms of $\mathcal{MVL}$ (denoted $\mathcal{A}$) are of the form v^{val} where $v \in \mathcal{V}$ and $val \in [\![0; \text{dom}(v)]\!]$.*

Definition 2 (Multi-valued logic program). *A $\mathcal{MVLP}$ is a set of $\mathcal{MVL}$ rules:*

$$\underbrace{v_0^{val_0}}_{head} \leftarrow \underbrace{v_1^{val_1} \wedge v_2^{val_2} \wedge v_3^{val_3} \wedge \dots \wedge v_m^{val_m}}_{body}$$

Definition 3 (Dynamic $\mathcal{MVLP}$). *Let $\mathcal{T} \subset \mathcal{V}$ and $\mathcal{F} \subset \mathcal{V}$ such that $\mathcal{F} = \mathcal{V} \setminus \mathcal{T}$. A $\mathcal{DMVLP}$ P is a $\mathcal{MVLP}$ such that $\forall R \in P, \text{var}(\text{head}(R)) \in \mathcal{T}$ and $\forall v^{val} \in \text{body}(R), v \in \mathcal{F}$.*

Definition 4 (Discrete state). *A discrete state s on $\mathcal{T}$ (resp. $\mathcal{F}$) of a $\mathcal{DMVLP}$ is a function from $\mathcal{T}$ (resp. $\mathcal{F}$) to $\mathbb{N}$. $\mathcal{S}^{\mathcal{T}}$ (resp. $\mathcal{S}^{\mathcal{F}}$) denote the set of all discrete states of $\mathcal{T}$ (resp. $\mathcal{F}$).*

Definition 5 (Transition). *A transition is a couple of states $(s, s') \in \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$.*

Definition 6 (Semantics). *A dynamical semantics is a function of $(\mathcal{DMVLP} \to (\mathcal{S}^{\mathcal{F}} \to \wp(\mathcal{S}^{\mathcal{T}}) \setminus \{\emptyset\}))$ where $\mathcal{DMVLP}$ is the set of $\mathcal{DMVLP}$s ($\wp$ is the power set symbol).*

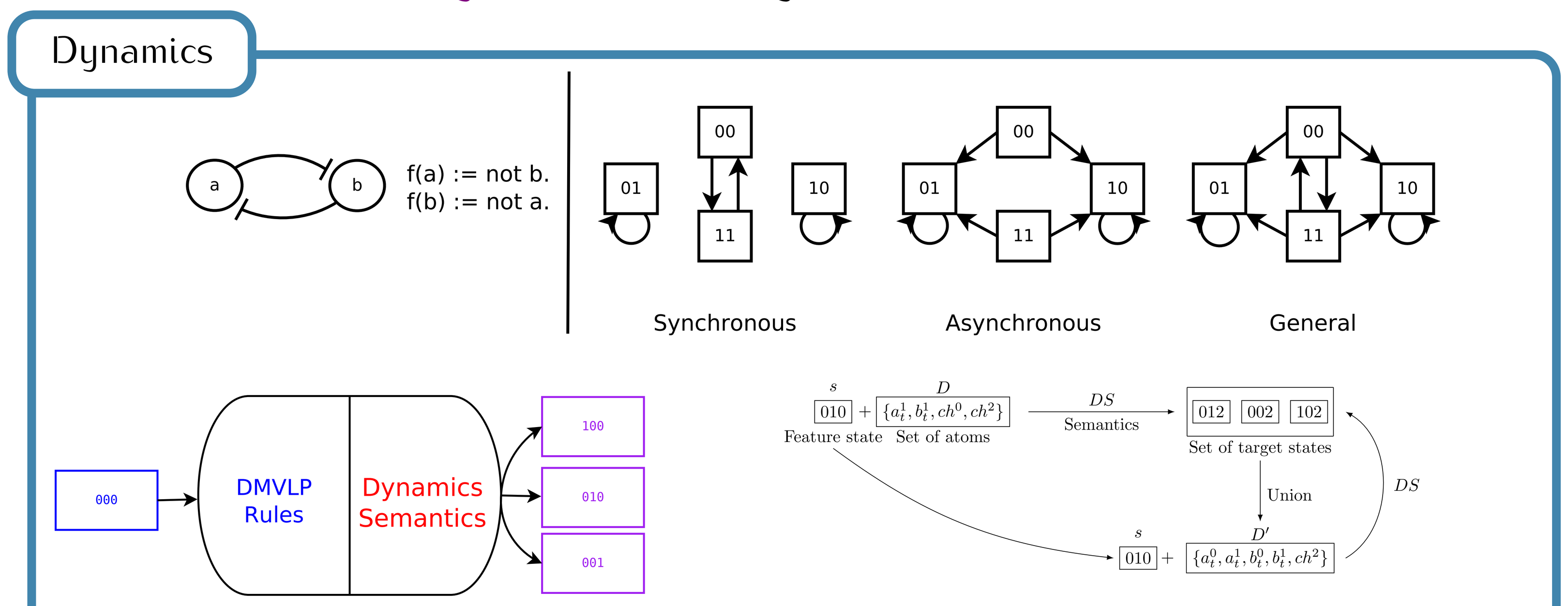
- R_1 dominates R_2, written $R_1 \geq R_2$ if $head(R_1) = head(R_2)$ and $body(_1) \subseteq body(_2)$.
- R matches $s \in \mathcal{S}^{\mathcal{F}}$, written $R \sqcap s$, if $body() \subseteq s$.
- R realizes the transition $(s, s') \in \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$, if $R \sqcap s, head(R) \in s'$.
- R conflicts with $T \subseteq \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$ when $\exists (s, s') \in T, \big(R \sqcap s \wedge \forall (s, s'') \in T, head(R) \notin s''\big)$.

Definition 7 (Suitable program). *Let $T \subseteq \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$. A $\mathcal{DMVLP}$ P is suitable for T when: P is complete, consistent with T, realizes T and $\forall R$ not conflicting with T, $\exists R' \in P$ s.t. $R \geq R'$. If in addition, $\forall R \in P$, all the rules R' belonging to a $\mathcal{MVLP}$ suitable for T are such that $R \geq R'$ implies $R' \geq R$ then P is unique, called optimal and denoted $P_{\mathcal{O}}(T)$.*

Problem: Dynamics Semantics

Semantics Decide the target states according to a $\mathcal{DMVLP}$ and a feature state.

Dynamics



A semantics that produce the same states, when being given the atoms of its own decision is pseudo-idempotent and is compatible with its transition optimal $\mathcal{DMVLP}$.

Definition 8 (Pseudo-idempotent Semantics). *Let DS be a dynamical semantics. DS is said pseudo-idempotent if, for all P a $\mathcal{DMVLP}$: $DS(P_{\mathcal{O}}(DS(P))) = DS(P)$.*

Algorithm: GULA

Definition 9 (Rule least specialization). *Let R be a $\mathcal{MVL}$ rule and $s \in \mathcal{S}^{\mathcal{F}}$ such that $R \sqcap s$. The least specialization of R by s according to $\mathcal{F}$ and $\mathcal{A}$ is:*

$$L_{spe}(R, s, \mathcal{A}, \mathcal{F}) := \{\text{head}(R) \leftarrow \text{body}(R) \cup \{v^{val}\} \mid v \in \mathcal{F} \wedge v^{val} \in \mathcal{A} \wedge v^{val} \notin s \wedge \forall val' \in \mathbb{N}, v^{val'} \notin \text{body}(R)\}.$$

$\forall T \subseteq \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$, we denote: $\text{first}(T) := \{s \in \mathcal{S}^{\mathcal{F}} \mid \exists (s_1, s_2) \in T, s_1 = s\}$.

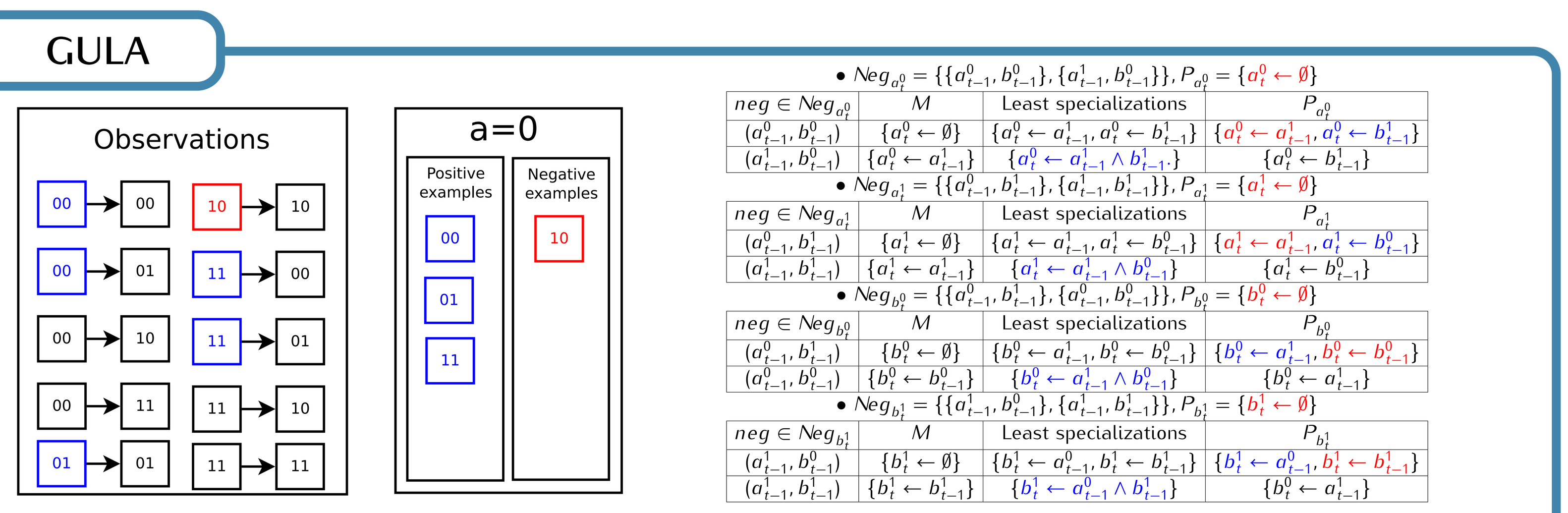
Definition 10 (Program least revision). *Let P be a $\mathcal{DMVLP}$, $s \in \mathcal{S}^{\mathcal{F}}$ and $T \subseteq \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$ such that $\text{first}(T) = \{s\}$. Let $R_P := \{R \in P \mid R$ conflicts with $T\}$. The least revision of P by T according to $\mathcal{A}$ and $\mathcal{F}$ is $L_{rev}(P, T, \mathcal{A}, \mathcal{F}) := (P \setminus R_P) \cup \bigcup_{R \in R_P} L_{spe}(R, s, \mathcal{A}, \mathcal{F})$.*

Algorithmic properties:

- $P_{\mathcal{O}}(\emptyset) = \{v^{val} \leftarrow \emptyset \mid v \in \mathcal{T} \wedge v^{val} \in \mathcal{A}\}$.
- Let $s \in \mathcal{S}^{\mathcal{F}}$ and $T, T' \subseteq \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$ such that $|\text{first}(T')| = 1 \wedge \text{first}(T) \cap \text{first}(T') = \emptyset$. $L_{rev}(P_{\mathcal{O}}(T), T', \mathcal{A}, \mathcal{F})$ is a $\mathcal{DMVLP}$ suitable for $T \cup T'$.
- If P is a $\mathcal{DMVLP}$ suitable for T, then $P_{\mathcal{O}}(T) = \{R \in P \mid \forall R' \in P, R' \geq R \implies R' = R\}$.

Idea: Starting from $P = P_{\mathcal{O}}(\emptyset)$ we group transitions by common feature state (T') and iteratively revise P using $L_{rev}(P, T', \mathcal{A}, \mathcal{F})$ and domination relation to obtain $P_{\mathcal{O}}(T)$.

GULA



We extract positive and negatives examples (feature states) of each target atom occurrence. A rule that matches a negative example conflict with the observations.

Learning From Any Semantics Using Constraints

Definition 11 (Constrained $\mathcal{DMVLP}$). *Let P' be a $\mathcal{DMVLP}$ on $\mathcal{A}^{\mathcal{F} \cup \mathcal{T}}_{\text{dom}}$, $\mathcal{F}$ and $\mathcal{T}$ two sets of variables, and ε a special variable with $\text{dom}(\varepsilon) = \{0, 1\}$ so that $\varepsilon \notin \mathcal{T} \cup \mathcal{F}$. A $\mathcal{CDMVLP}$ P is a $\mathcal{MVLP}$ such that $P = P' \cup \{R \in \mathcal{MVL} \mid \text{head}(R) = \varepsilon^1 \wedge \forall v^{val} \in \text{body}(R), v \in \mathcal{F} \cup \mathcal{T}\}$. A rule R such that $\text{head}(R) = \varepsilon^1$ and $\forall v^{val} \in \text{body}(R), v \in \mathcal{F} \cup \mathcal{T}$ is called a $\mathcal{MVL}$ constraint.*

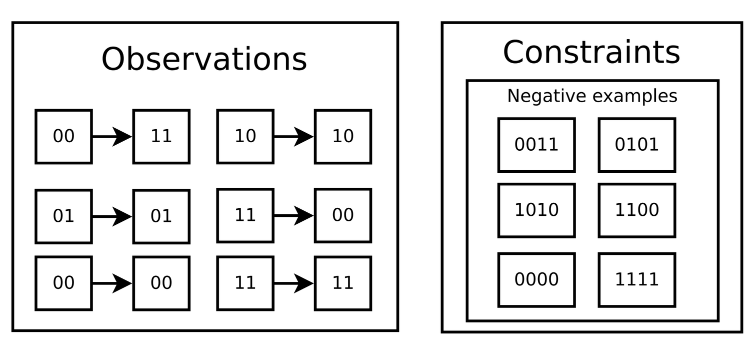
Definition 12 (Constraint-transition matching). *Let $(s, s') \in \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$. The constraint C matches (s, s'), written $C \sqcap (s, s')$, iff $\text{body}(C) \subseteq s \cup s'$.*

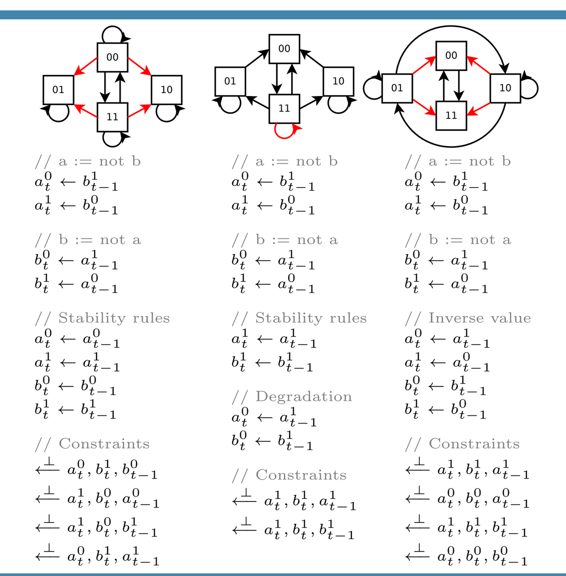
Definition 13 (Suitable and optimal constraints). *Let $T \subseteq \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$. A set of $\mathcal{MVL}$ constraints SC is suitable for T when: SC is consistent with T, complete with T and for all constraints C not conflicting with T, there exists $C' \in P$ such that $C' \geq C$. If in addition, for all $C \in SC$, all the constraint rules C' belonging to a set of constraints suitable for T are such that $C' \geq C$ implies $C \geq C'$, then SC is called optimal, is unique and denoted $C_{\mathcal{O}}(T)$.*

Definition 14 (Synchronous constrained Semantics). *The synchronous constrained semantics $\mathcal{T}_{syn-c}$ is defined by:*

$$\mathcal{T}_{syn-c} : P \mapsto \{(s, s') \in \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}} \mid s' \subseteq \text{Conclusions}(s, P) \wedge \nexists C \in P, \text{head}(C) = \varepsilon^1 \wedge C \sqcap (s, s')\}$$

Synchronizer





Negative examples of constraints are the observed transitions. GULA can be used to learn constraint with this simple trick.

Let $T \subseteq \mathcal{S}^{\mathcal{F}} \times \mathcal{S}^{\mathcal{T}}$, it holds that $T = \mathcal{T}_{syn-c}(P_{\mathcal{O}}(T) \cup C_{\mathcal{O}}(T))$, i.e., any semantics is captured.

Contributions

- **Previous works:** Synchronous deterministic transitions only [1-3].
- **Novelty:** Learn from any memory-less discrete dynamics semantics.
- **Application:** semantic choice, which has an important meaning for the one who try to model a system, can now be done a posteriori. The rules can explain local interactions and constraint are hints of semantics behaviors.
- **Weakness:** current complete method is too costly/sensitive to deal with real system.
- **Outlook:** development of heuristic approach ($\mathcal{WDMVLP}$, PRIDE) to tackle real data and tools (see other poster) to extract knowledge from the learned models.
- The source code is available as open source on Github. See QR-code ⟶

[1] Tony Ribeiro, Sophie Tourret, Maxime Folschette, Morgan Magnin, Domenico Borzacchiello, Francisco Chinesta, Olivier Roux, Katsumi Inoue. Inductive Learning from State Transitions over Continuous Domains, The 27th International Conference on Inductive Logic Programming, (ILP 2017), Orléans, France.

[2] Tony Ribeiro, Katsumi Inoue. Learning Prime Implicant Conditions From Interpretation Transition, Inductive Logic Programming: Revised and Selected Papers from the 24th International Conference on Inductive Logic Programming, (ILP 2014), pages 108-125, Nancy, France. *Lecture Notes in Artificial Intelligence*, Springer.

[3] Katsumi Inoue, Tony Ribeiro, Chiaki Sakama. Learning from Interpretation Transition, *Machine Learning Journal*, volume 94, issue 1, pages 51-79.