Theory Synthesis based on Experience

Yannick Chevalier

Institut de Recherche en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

Theory Synthesis based on Experience

Yannick Chevalier*
IRIT, Toulouse University, CNRS, INP, UT3, Toulouse, France
Yannick.Chevalier@irit.fr

* contact author

October 3rd, 2022

Technical report No. IRIT/RR–2022–08–FR
(version 1)

Institut de Recherche en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

Theory Synthesis based on Experience

Yannick Chevalier*
IRIT, Toulouse University, CNRS, INP, UT3, Toulouse, France
Yannick.Chevalier@irit.fr

* contact author

October 3rd, 2022

Abstract. We present in this paper a novel approach to learning that produces practical and conceptual knowledge. The former aims at detecting changes or new behaviour in observations, a trait of anomaly detection systems, and that practical knowledge can be updated based on an Angluin-like property. The second one aims at enabling communication with an expert user by producing a first-order logic model of the world observed.

This construction fills the gap in previous work by Achourioti and van Lanbalgem were inverse systems of first-order logic models are employed to formalize Kant's transcendental logic.

More precisely learning is modeled by algebraic morphisms on algebraic lattices. Composition of morphisms is denoted by first-order terms interpreted as intensional tables or constraints on tables. Data Exchange Systems (DXS) are set of terms describing databases storing information about observations and observed constraints. Practical knowledge is a valuation of a DXS and the associated skill is to decide whether a new event is conform to past events, a new behaviour or an anomaly. When the decision is wrong, the DXS is updated according to whether an anomaly or a new normal behaviour was wrongly assessed.

These DXS are enriched with sets of terms representing tables of interest or generic constraint, resulting in cognitive states. A first-order model is constructed for each compact cognitive state and contains the set of predicates that are sensible given the observations and the cognitive state. We finally prove that these models form an inverse system of models whose limit is the starting point of the formalisation of Kant's Transcendental Logic.

Keywords: Algebraic domains ; Geometric Logic ; Knowledge Representation; Learning

Technical report No. IRIT/RR–2022–08–FR
(version 1)

Institut de Recherche en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

Synthèse de Théories basée sur l'Expérience

Yannick Chevalier*
IRIT, Université de Toulouse, CNRS, INP, UT3, Toulouse, France
Yannick.Chevalier@irit.fr

* contact author

3 octobre 2022

Résumé. We present in this paper a novel approach to learning that produces practical and conceptual knowledge. The former aims at detecting changes or new behaviour in observations, a trait of anomaly detection systems, and that practical knowledge can be updated based on an Angluin-like property. The second one aims at enabling communication with an expert user by producing a first-order logic model of the world observed.

This construction fills the gap in previous work by Achourioti and van Lanbalgem were inverse systems of first-order logic models are employed to formalize Kant's transcendental logic.

More precisely learning is modeled by algebraic morphisms on algebraic lattices. Composition of morphisms is denoted by first-order terms interpreted as intensional tables or constraints on tables. Data Exchange Systems (DXS) are set of terms describing databases storing information about observations and observed constraints. Practical knowledge is a valuation of a DXS and the associated skill is to decide whether a new event is conform to past events, a new behaviour or an anomaly. When the decision is wrong, the DXS is updated according to whether an anomaly or a new normal behaviour was wrongly assessed.

These DXS are enriched with sets of terms representing tables of interest or generic constraint, resulting in cognitive states. A first-order model is constructed for each compact cognitive state and contains the set of predicates that are sensible given the observations and the cognitive state. We finally prove that these models form an inverse system of models whose limit is the starting point of the formalisation of Kant's Transcendental Logic.

Mots-clés : Algebraic domains ; Geometric Logic ; Knowledge Representation; Learning

Rapport technique N° IRIT/RR–2022–08–FR
(version 1)

1 Introduction

Background and motivations. Learning and knowledge are introduced along seemingly independent approaches. In a symbolic setting, *learning* was first defined [19] as improving a procedure guessing the membership of a word in a regular language. Angluin's theory [8] focus on learning the language given membership examples and counter-examples. PAC learning [18] generalised further the problem by assuming a probability of failing to learn, and when succeeding in learning a possibility of error in answering queries.

The definition of practical knowledge in this paper stems from the need to formalise an absence of knowledge, *e.g.*, when analysing pseudo-random sequences. These are defined in [3] as sequences such no Turing Machine is able to compute in polynomial time a meaningful prediction of the next bit given a sequence of bits already produced. This approach is extended in computational cryptography where having *zero knowledge* about an object represented with a string s is modeled as having no couple of Turing Machine (TM) such that the second one is able to determine whether the first one had s or a random string on its input tape [32]. To introduce the vocabulary of this paper, an *object* is formed by the first TM, and the second TM computes a *representation*. Absence of knowledge is defined as the inability to compute any meaning representation. As a contrapositive, knowing something is being able to compute a meaningful representation.

More recently and more directly related to this paper it was proven [11] that symbolic first-order logic techniques can be employed to model and decide computational cryptography lack of knowledge by *static equivalence* [21]. In this symbolic setting, agents' knowledge about a term describing a sequence of messages is specified by a set of equations satisfied only by possible instances of these messages. Under some additional hypotheses it is possible [35] to compute from a generic term a set of tests that entirely determines membership in the set.

Finally *formal concept analysis* (FCA) [6, 26] embeds objects and attributes into a lattice to form concepts of similar objects with respect to their attributes. The partial grouping of objects per similarity can be modeled with a partial equivalence relation that naturally leads to algebraic lattices [2]. *Conceptual knowledge* is the extension of concepts to predicates of arbitrary arity, while classes of indistinguishable object terms (according to the predicates at hand) form a domain on which these predicates are interpreted.

Approach. In response to Empiricists such as Hume, Kant provided an answer to the problem of causation based on the analysis of human thought process [14]. Though Frege's formalisation of first-order logic [12] seemed to enlighten Kant's formalism, it was objected in [5] that this formalisation over-simplified Kant's work by reducing the act of the mind called *cognition* to *deduction*, a mere syntactic manipulation. More recently, inverse systems of first-order logic models were proposed [30] as a basis to formalise Kant's analysis. A difficulty in any tentative formalisation is that according to Kant logic is based on cognitions, that these cognitions define a semantics, and that this semantics determines the syntax, a departure from more modern descriptions of logic.

Beyond philosophy and metaphysical considerations, puzzling out Kant's analysis is interesting to the modern computer scientist as it promises a natural description of observations under the guidance of primitive *categories* which in addition to causality encompass temporality and space. This approach is highly relevant *e.g.*, in *anomaly detection* in which a usable system must be able to describe suspicious behaviour to network experts in a high-level language [25].

This paper extends the proposal in [30] as follows. Cognitions are defined as algebraic morphisms between algebraic lattices. Their compositions is denoted with first-order terms whose interpretation in algebraic lattices is updated to fit observations. A single comparison predicate is added to create an *observation* model. These terms are separated into *object* and *representation* terms. Two subsets of presentation and object terms considered for further examination determine respectively a set of predicates as conjunction over the observation model and a domain as a set of partial equivalence classes of object terms *wrt* these predicates. We prove these models form an inverse system of models.

Limitations. The construction presented in this paper coincides with the one in [30] when the observation of the world is fixed. This can be explained by the focus in [30] on the construction of a model in which the cognitions and the logic can be characterised as *geometric logic* [29] . There are differences in our construction *wrt* Kant's model of cognition as presented in [5]: the choice of remembering past observations rather than reconstructing them, and the lack of focus on the temporal relation between events.

Organisation. The choice of algebraic lattices and morphisms to denote respectively knowledge and learning is justified in Section Section 2, and the corresponding **aLAT**$_!$ category is introduced with its salient properties. The learning process is defined in Section Section 3. The target of this first learning process is *practical knowledge*, introduced together with a limited formalisation of *skills* in Section Section 4. It is proved that the skill can be improved through learning, an Angluin-like property. Section Section 5 introduces cognitive states and Conceptual knowledge as first-order models on compact cognitive states before being extended to all stable cognitive states in Section Section 6.

Future works are presented in Section Section 7.

2 Modeling Learning

2.1 Informal considerations

Per the Oxford dictionary, *learning* is the acquisition of knowledge or skills through *study*, *experience*, or *being taught*.

Without going into the definitions of knowledge and skills it can be infered that they increase through the three stated activities. Let d, d' be datasets representing observations of the world, and assume that $d \subseteq d'$. Then a learning algorithm f applied on d' should return a better result than when applied on d, something we denote with $f(d) \sqsubseteq f(d')$. We infer from this comparison that both the input and output of the function must be posets and that f must be a monotonic between these posets. Reasoning along the same lines, we conclude that any learning function must be monotonic *wrt* studying (*cognitions*) and being taught (user-given information).

Independently, Kant defines logic as the rules of discurse that enable the construction of truth from observations. A central tenet in his analysis is the always present possibility of reconciling different positions as long as they were established through the proper rules. This reconciliation process is presented in [30] through the *unity of self* that must be preserved during the establishment of any proper theory of the world. Mathematically speaking, if both d, d' are datasets representing different observations of the world and f is a learning algorithm, denoting $\sqcup$ the amalgamation of the experiences and of their results we have $f(d \sqcup d') = f(d) \sqcup f(d')$. This implies *inter-alia* that the codomain of f is a join-semilattice. It is then natural to also assume that if $d_\perp$ represents the minimum

experience, then $f(d_\perp)$ should also be the minimum value in the codomain of f.

Another postulate is that the theories properly constructed from experience should converge towards an ideal description of the world. It was already noted in [27] that this postulate implies the existence of a Stone duality [20] theorem between the world as a model and the theory constructed. To construct a meaningful theory one has to assume that the observed world is not random and thus that not all sequences of events are possible. Thus the possible observations of the world are bounded by a set d of possible sequences. Then for all sequences of sets of observations $(d_n)_{n\in\mathbb{N}}$ that converges towards d, *i.e.*, such that $\cup_{n\in\mathbb{N}} d_n = d$, we should also have $\bigsqcup_{n\in\mathbb{N}} f(d_n) = f(d)$, even if the latter can only be asserted to exist and not computed explicitely. This implies that the limits $\cup_{n\in\mathbb{N}} d_n$ and $\bigsqcup_{n\in\mathbb{N}} f(d_n)$ exist if $(d_n)_{n\in\mathbb{N}}$ is an increasing sequence of observations. Since both the domain and codomains are lattices, this implies that they are actually *complete lattices*, and that f is *Scott-continuous*.

2.2 Model for machine learning

We now present the formal setting, with *well-known* referencing definitions and properties in [28].

Definitions. Let $(P, \sqsubseteq)$ be a poset. Given $x \in P$ we denote $\downarrow x = \{y \in P \,|\, y \sqsubseteq x)\}$. A subset $X \subseteq P$ is an ideal whenever $x \in X$ and $y \sqsubseteq x$ imply $y \in X$. The set of ideals of P ordered by inclusion is denoted $\mathrm{Idl}(P)$. It is well-known that since P is a poset, $\mathrm{Idl}(P)$ is a complete lattice: the infimum and supremum all subsets of $X \subseteq \mathrm{Idl}(P)$ are defined, and denoted respectively $\sqcap X$ and $\sqcup X$.

An element $x \in \mathrm{Idl}(P)$ is *compact* whenever for all sets X, $x \sqsubseteq \sqcup X$ implies there exists a finite $Y \subseteq X$ such that $x \sqsubseteq \sqcup Y$. Since $\mathrm{Idl}(P)$ is a complete lattice it is bounded complete: if $x, y \sqsubseteq z$ then $x \sqcup y \sqsubset z$. A subset X of P is directed whenever for all $x, y \in X$, there exists $z \in X$ such that $x \sqsubseteq z$ and $y \sqsubseteq z$. It is well-known that since P is a poset, the compact elements $K(\mathrm{Idl}(P))$ of $\mathrm{Idl}(P)$ are (up to isomorphism) the elements of P. Given $x \in \mathrm{Idl}(P)$ denote $x\downarrow = K(\mathrm{Idl}(P)) \cap \downarrow x$. It is well-known that this set is directed, and that $x = \sqcup x\downarrow$ for all $x \in \mathrm{Idl}(P)$: $K(\mathrm{Idl}(P))$ is a *base* of $\mathrm{Idl}(P)$, thus $\mathrm{Idl}(P)$ is an *algebraic lattice*.

Traces, Observations, and Object Domains. Let A be an alphabet of events. A *trace* is a word in A^*. During the learning phase, a set of traces L is availabe for learning. For $u, v \in A^*$ denoting $u \sqsubseteq v$ if u is a prefix of v, we note that whenever v is observed and $u \sqsubseteq v$ then u also is observed. Thus the set of effectively observed traces in the course of learning is $\downarrow L$, that is the *ideal* defined by L, from now on called an *observation*. Since $(A^*, \sqsubset)$ is a poset, $\mathrm{Idl}(A^*)$ is an algebraic lattice. The compact ideals of $\mathrm{Idl}(A^*)$ are generated by finite subsets of A^*. We denote a compact ideal I with $\Sigma_{i=1}^n u_i$ if $\{u_1, \ldots, u_n\}$ is the intersection of all subsets $S \subseteq A^*$ such that $\downarrow S = I$. Given an alphabet A $\mathrm{Idl}(A^*)$ is denoted T_A and called the *object domain of* A.

The **aLAT**$_!$ *category.* We consider in this paper *domains* which are *algebraic lattices*. It is well-known they are always complete. The bottom (*resp.* top) of a domain A is $\sqcup\emptyset = \perp_A$ (*resp.* $\sqcap\emptyset = \top_A$). We consider morphisms between domains that are:

- *Continuous:* $f(\sqcup X) = \sqcup_{x\in X} f(x)$ for every $X \subseteq A$);
- *Strict:* If $f : A \to B$, then $f(\perp_A) = \perp_B$.

A domain morphism mapping compact elements to compact elements is *algebraic*. A *learning function* is an algebraic morphism. The objects of the **aLAT**$_!$ category are the domains and its morphisms are the strict continuous morphisms.

Example 1. Any finite lattice is a domain. Thus for example the usual boolean lattice $\mathbb{B} = \{0, 1\}$ with $1 \sqsubset 0$ is a domain. The Belnap four-valued lattice [23] $\{U, T, F, C\}$ (unknown, true, false, contradictory) with $U \sqsubset T, F$ and $T, F \sqsubset C$ is also a domain. More generally any finite lattice, including concept lattice [26] employed to classify data hierarchically, is a domain. Another generalisation of Belnap's four-valued lattice is to consider any unordered set S and equip it with a top and a bottom element. The resulting ordered set $S \cup \{\top, \bot\}$ is a domain. Finally the set of subsets of $\mathbb{N}$ equiped with the union operation and the inclusion ordering is a domain whose compact elements are the finite subsets of $\mathbb{N}$.

Example 2. Let D_{spr} be the domain $\mathbb{R}^+ \cup \{+\infty\}$ equiped with $x \sqcup y = \min(x, y)$. The minimum of that domain is $+\infty$ and its top element is 0. The minimum is the only compact element. Equiped with D_{spr} it is possible to memorize the record time in a sequence of sprint races.

Lifting. It is convenient to define functions the alphabet A, or at least on finite words. Such definitions, provided the codomain is a domain, can be canonically extended into a morphism.

Definition 1. Let A be an alphabet, D be a domain, and $f : A^* \to D$ be such that for all $u, v \in A^*$, $u \sqsubseteq v$ implies $f(u) \sqsubseteq f(v)$. The *lifting of* f is denoted $\hat{f} : T_A \to D$ and is defined by:

$$\hat{f}(u) = \bigsqcup_{v \in \downarrow u} \bigsqcup_{\substack{v = \Sigma_{w \in B} w \\ w \in B}} f(w)$$

The definition of $hat f$ ensures that all limits commute with its application.

Proposition 1. *Let A be an alphabet, D be a domain, and $f : A^* \to D$ be such that for all $u, v \in A^*$, $u \sqsubseteq v$ implies $f(u) \sqsubseteq f(v)$. Then $\hat{f} \in [T_A \to D]$ is also algebraic.*

2.3 Properties of the aLAT$_!$ category

It is well-known that the **aLAT**$_!$ category is cartesian closed. [6] In particular its exponential is an algebraic lattice. The *support* of a function $f : D \to E$ is the minimal subset $S \subseteq D$ such that $x \notin S$ implies $f(x) = \bot_E$. It is well-known that the compact elements of $[D \to E]$ are the functions with a finite support S such that $f(x)$ is compact for all $x \in S$.

Proposition 2. *If D, E are objects in* **aLAT**$_!$ *then $[D \to E]$ also in* **aLAT**$_!$.

Proof. By Prop. 4.1.5 of [27]: algebraic lattices are continuous lattices. □

Example 3. In particular a database table whose elements are in a domain and indexed by keys in a domain can be encoded as a function between these domains. To be admissible, the construction of a database must be an algebraic morphism and thus map observations to finite tables.

In particular we use without further reference that the product $\Pi_{n \in \mathbb{N}} D_n$ is in

aLAT$_!$ if $D_n \in$ **aLAT**$_!$ for all $n \in \mathbb{N}$..

2.4 Split functions

Observations are analysed in a loop involving first the discovery of properties of events, then the classification of events according to these properties. This classification yields a denumerable set of classes that can be analysed independently. This possibility of finer-grained analysis justifies the introduction of specific classification morphisms, the *split functions*.

Example 4. Continuing the sprint example, it is natural to classify the humans running according to their speed, *e.g.*, define athlete as someone able to run in less than 12s.

This classification should be complete, *i.e.*, encompass all the events seen so far. It is also arbitrary, in the sense that assuming an object domain T_S, a *choice function* c is just any function $S \to \mathbb{N}$. The *split function for* c maps each event e to the $c(e)$th copy of S. It is denoted Split_c and maps an object $u \in T_S$ to an element $\mathrm{Split}_c(u) = (u_n)_{n\in\mathbb{N}} \in \Pi_{n\in\mathbb{N}} T_S$. The notation $(u_n)_{n\in\mathbb{N}}$ is defined on events and the Split_c function is lifted to elements of T_S with Prop. 1.

Proposition 3. *Let T_S be an object domain, and c be a choice function on S. Then Split_c is algebraic.*

Proof. First since **aLAT**$_!$ is cartesian closed $\Pi_{n\in\mathbb{N}} T_S$ is in **aLAT**$_!$. The Split_c morphism is by construction strict and continuous, we have to prove it is algebraic. The compact elements of its codomain are the sequences of finite support such that all the values are compact. Let $u = (u_n)_{n\in\mathbb{N}} \in \Pi_{n\in\mathbb{N}} T_S = \mathrm{Split}_c(v)$. If v is compact then it is an observation, a finite set of traces, and thus there is only a finite number of events occuring in v. Each event has one class, thus $\mathrm{Split}_c(v)$ is of finite support. For all $n \in \mathbb{N}$ the value u_n is an observation and thus is compact. Thus if v is compact then $\mathrm{Split}_c(v)$ is compact. □

Example 5. The domain D_{spr} of Ex. 2 is a representation domain. A function that extracts from each event its runner and run time can be lifted into an algebraic morphism mapping an observation u to another observation u'. One can compute a representation of u'–and therefore of u–with a table associating to persons their record time. A representation containing the minimum value in that table can then be computed. A piece of knowledge that can be extracted from this representation of the observation u is that all humans run 100m in at least that minimum value.

3 Learning

Given our primary interest on anomaly detection systems we borrow from [34] the naming conventions on filters and *Data Exchange Systems*. Summarily this work on the resolution of simple Data Exchange (DX) problems to fill tables in a database and on *filters* (to be understood as filters in *dynamical system* analysis, not as the topological construction) to learn integrity contraints. This generalisation follows the approach to DX of [9] in which rules are modeled as morphisms between objects that represent tables. In contrast with that work we are not interested in learning complex diagrams, and focus on the values computed in

acyclic diagrams with one source. Accordingly a database is modeled with a set of terms, each term denoting either the composition of functions filling a table (an object term) or a filter (a representation term).

3.1 Notations

We let $D_0, D_1, \ldots, D_n$ be objects in **aLAT**$_!$. The domain D_0 plays the particular role of being an object domain which is the source of the observations. Some of these are *object domains*, the other are called *representation domains*. We consider a set F of algebraic morphisms $f_1, \ldots, f_m$ between these domains, with $f \in F$ of arity k_f and:

$$f : D_{\alpha_f(1)} \times \cdots \times D_{\alpha_f(k_f)} \to D_{\alpha_f(0)}$$

for a mapping $\alpha_f : \{0, \ldots, k_f\} \to \{0, \ldots, n\}$. We also let S be a set of n_s split functions $\{\text{Split}_{c_i}\}_{1 \leq i \leq n_s}$ over objects domains $D_{\alpha_{c_i}}$ for $\alpha_{c_i} \in \{1, \ldots, n\}$. A function is a *representation* whenever its codomain is a representation domain, and an object function otherwise.

First-order signature. A sort τ_i is associated with each domain D_i, for $0 \leq i \leq n$. The sorted first-order functional signature $\Sigma_{F,S}$ comprises

1. The set of function symbols $\mathrm{f}_1, \ldots, \mathrm{f}_m$ with sorts
$$\mathrm{f} : \tau_{\alpha_f(1)} \times \cdots \times \tau_{\alpha_f(k_f)} \to \tau_{\alpha_f(0)}$$
2. A unique constant "X":τ_0 denoting the object domain under analysis;
3. For each $\text{Split}_c : D_{\alpha_c)} \to \Pi_{n \in \mathbb{N}} D_{\alpha_c} \in S$ and each $n \in \mathbb{N}$, a function symbol $\mathrm{c}^n : \tau_{\alpha_c} \to \tau_{\alpha_c}$.

The set of *ground terms over* $\Sigma_{F,S}$ is as usual the least set $\mathrm{T}(\Sigma_{F,S})$ such that:

- "X":$\tau_0 \in \mathrm{T}(\Sigma)$;
- if $t_1{:}\tau_1, \ldots, t_{k_f}{:}\tau_{k_f} \in \mathrm{T}(\Sigma)$ and $\mathrm{f} : \tau_{\alpha_f(1)} \times \cdots \times \tau_{\alpha_f(k_f)} \to \tau_{\alpha_f(0)}$ then $f(t_1, \ldots, t_n)$: $\tau_{\alpha_f(0)} \in \mathrm{T}(\Sigma)$;
- If $t : \tau \in \mathrm{T}(\Sigma)$, $t \neq \mathrm{c}^n(t')$ for some $n \in \mathbb{N}$, then for all $m \in \mathbb{N}$, $\mathrm{c}^m(t) \in \mathrm{T}(\Sigma)$.

The last rule prohibits the application of a split function on a class that is the result of the application of the same split function. In most cases the specific set of functions is not relevant to the analysis, and we denote simply Σ a sorted first-order functional signature as defined above.

Positions and subterms. A *position* is a finite sequence of integers, with ϵ denoting the empty sequence. Positions in and subterms of a term t are defined recursively as follows:

- t is a subterm of t at position ϵ;
- If $\mathrm{f}(t_1, \ldots, t_n)$ is a subterm at position p in t, then each t_i is a subterm at position $p \cdot i$ in t.

Given a term t we denote $\text{Sub}(t)$ its set of subterms.

Representation and object terms. The terms $t : \tau$ such that $\tau = \alpha_f(0)$ for a representation function f are called *(ground) representation terms*. Otherwise they are called *(ground) object terms*. Given a set of terms S we denote $\text{Repr}(S)$ the subset of representation terms in S.

Interpretation of $u \in D_0$. The interpretation of a function symbol f is denoted [f] and is f. The interpretation of a term t over $u \in D_0$ is denoted $[t]_u$ and defined inductively on ground terms as expected:

$$\begin{cases} [\text{"X"}]_u = u \\ [\text{f}(t_1, \ldots, t_n)]_u = f([t_1]_u, \ldots, [t_n]_u) \\ [\text{c}^n(t)]_u = \text{Split}_c([t]_u)(n) \end{cases}$$

If t is a representation term we distinguish its interpretation on u by calling it a *value* and denoting it $\text{Val}_u(t))$. The codomain of $\text{Val}_u(t))$ is denoted $\text{coDom}(t)$.

3.2 Data Exchange Systems (DXS)

A DXS is a set of terms over a signature $\Sigma = \Sigma_{F,S}$. with a few conditions that we explicit in this section.

Definition 2. (Data Exchange System) Let Σ be a signature. A Σ-*Data Exchange System* (Σ-*DXS*) is a non-empty set of ground terms $S \subseteq \text{T}(\Sigma)$ such that:

- "X" $\in S$;
- Each term $t \in S$ is the result of the application of a function in Σ on a finite subset of S;
- If $\text{c}^n(t) \in S$, then for all $m \in \mathbb{N}$, $\text{c}^m(t) \in S$.

We denote $\mathcal{D}(\Sigma)$ the set of Σ-Data Exchange Systems.

The *initial DXS* $\bot_{\mathcal{D}(\Sigma)}$ is {"X"}. Given two possible DXS S,S' we denote $S \sqsubseteq S'$ if $S \subseteq S'$ as sets. $\mathcal{D}(\Sigma)$ with a poset with the ordering $\sqsubset$ and the join operation $\Delta \bigsqcup \Delta' = \Delta \cup \Delta'$. Let us prove it is a domain. First we note it has a maximal element, $\text{T}(\Sigma)$ and a minimal element, the initial DXS. Per definition they are a closed subset of the powerset of the set of terms, and thus lattices. It remains to prove they are algebraic.

Consider a *cognition* Add_t that adds a term t to a DXS, or if t is one of the result of a split function c on a term t, adds simultaneously all the possible results $c^n(t)$. Let $\mathcal{C}_\Sigma$ be the set of all these cognitions. The constraint on the addition of terms in the definition of DXS, namely that before a term is added all its subterms must be present, naturally orders the set $\mathcal{C}_\Sigma$ with $\text{Add}_t \subseteq \text{Add}_{t'}$ whenever t (or one of the cases if t denotes the application of a split function) is a subterm of t'. Reusing the discussion on the construction of object domains, it is then natural to consider the algebraic lattice $L_{\mathcal{C}_\Sigma}$, or more simply $L_\mathcal{C}$ if Σ is clear from the context, of ideals of $\mathcal{C}_\Sigma$. Given that the elements of $L_\mathcal{C}$ are exactly the allowed constructions for DXS, we have the following proposition, which in turn entails that $\mathcal{D}(\Sigma)$ is also an algebraic lattice.

Proposition 4. *$L_{\mathcal{C}_\Sigma}$ is isomorphic (as a lattice) to $\mathcal{D}(\Sigma)$.*

As a corollary the compact DXS are the images of finite sets of cognitions in $\mathcal{C}_\Sigma$.

Since DXS are domains it is possible to define morphisms on DXS domains to extend the signature with new morphisms or split functions obtained either

through user-interaction (a case of *teaching*) or through the system's own cognitions.

Proposition 5. *(Teaching) Let Σ, Σ' be two representation signatures such that $\Sigma \subseteq \Sigma'$. Then the identity injection: $\iota : \mathcal{D}(\Sigma) \to \mathcal{D}(\Sigma')$ is an algebraic morphism.*

The morphism on signatures (as denumerable sets of symbols) adding a new function is clearly algebraic. It shall be noted that the construction presented so far to define *learning* encompasses experience through the observations, studying through the cognitions, and being taught through signature extensions. This delineation is informal: On the one hand, concepts learned from a preliminary analysis can lead to the addition of a new split function, and the detection of a functional relation between terms can lead to the addition of a new function to the signature without any interaction or "being taught". On the other hand the role of *categories* in Kant's analysis, especially in relation with the table of judgements, is one functions that *hold together* several objects to produce a new one, and some of these categories are assumed to be innate. Functions in the signatures are those categories that hold together the objects of perception regardless of whether they are innate, taught, or infered by cognitions.

4 Practical Knowledge and Skills

Knowledge has already been defined many times in many different settings. It seems nonetheless appropriate to add two new definitions. In this section *Practical Knowledge is the valuation of representation terms.* The *skill* steming from that knowledge is the ability to detect anomalies, a process called *monitoring*. In essence, *algebraicity* entails that the set of all possible observations can be approximated by finite observations; *Continuity* then imply the convergence of the valuation learned to that of the ideal representation. The system is self-correctible when given labeled observations.

4.1 Practical Knowledge

First we prove that the valuation of terms in a domain is continuous as it is the composition of continuous functions. Let Σ be a signature, S be a DXS in $\mathcal{D}(\Sigma)$, D be a domain, u be an observation, and t be a term. These notations are decorated reasonably, and D_0 is the domain on which "X" is interpreted.

Lemma 1. *The mapping $u \in D_0 \mapsto [t]_u$ is an algebraic morphism.*

Proof. By contradiction assume the set Ω of terms t such that the mapping $u \in D_0 \mapsto [t]_u$ is not continuous is not empty. Let t be minimal for the well-founded subterm relation in Ω. We must have $t \neq$ "X". If $t = f(t_1, \ldots, t_n)$, then by minimality of t the functions $u \mapsto [t_i]_u$ are continuous, and thus by function composition the function $u \mapsto [t]_u$ must be continuous as both morphisms and split functions are continuous, a contradiction. Algebraic functions map compact to compact, thus their finite compositions are algebraic. □

In the following proposition, since the mapping to each coordinate is continuous by Lemma 1, the mapping to the product of interpretations is continuous.

Proposition 6.

$$\begin{array}{rccl} \mathrm{Val}^S: & D_0 & \to & \Pi_{t\in \mathrm{Repr(T(\Sigma))}}\,\mathrm{coDom}(t) \\ & u & \mapsto & \Pi_{t\in \mathrm{Repr(T(\Sigma))}} d_t \textit{ with} \\ & & & d_t = \begin{cases} \mathrm{Val}_u(t) & \textit{if } t \in S \\ \bot_{\mathrm{coDom}(t)} & \textit{Otherwise} \end{cases} \end{array}$$

is a morphism. It is algebraic if S is compact.

Proof. We have already seen that the codomain $\Pi_{t\in \mathrm{Repr(T(\Sigma))}} D(t)$ is in **aLAT**$_!$. By Lemma 1 the projections to each coordinate are continuous. Thus by the universal property of the product topology this function is continuous. Also it clearly maps ϵ to the product of the minimal elements, which is the minimal element of the product.

It remains to prove it is algebraic when S is compact. Assume this is the case and let $u \in D_0$ be an observation. Since u is compact it contains a finite number of events. Since S is compact, it contains only a finite number of non-split terms, and a finite number of splits (each of which adding a denumerable number of terms). By Prop. 3 and since u is compact for each split only a finite number of terms have a non-bottom valuation.

Thus if S is compact, all but finitely elements of the product have the $\bot$ valuation, the remaining ones having a compact value since the interpretation of each term is compact by Lemma 1. □

The valuation is also continuous if the DXS changes with a fix world u.

Proposition 7.

$$\begin{array}{rccl} \mathrm{Val}_u: & \mathcal{D}(\Sigma) & \to & \Pi_{t\in \mathrm{Repr(T(\Sigma))}}\,\mathrm{coDom}(t)) \\ & S & \mapsto & \mathrm{Val}_u^S \end{array}$$

is a morphism. It is algebraic if u is compact.

Proof. "X" is not a representation term and thus Val_u maps the bottom element of $\mathcal{D}(\Sigma)$ to the bottom element of the product. Its continuity again is derived from the fact that the mapping to each coordinate t is continuous as it can only change once from $\bot_{D(t)}$ to $\mathrm{Val}^{D(t)}([\![t]\!]_u)$.

When u is compact, the fact that it maps compact DXS to compact elements of the product is already proved in the proof of Prop 6. □

A function continuous in each argument being continuous on the product, we obtain the following theorem that characterizes our learning approach.

Theorem 1.

$$\begin{array}{rccl} \mathrm{Val}: & D_0 \times \mathcal{D}(\Sigma) & \to & \Pi_{t\in \mathrm{Repr(T(\Sigma))}}\,\mathrm{coDom}(t)) \\ & (u, S) & \mapsto & \Pi_{t\in \mathrm{Repr(T(\Sigma))}}\,\mathrm{Val}_u^S(t) \end{array}$$

is algebraic and continuous.

Proof. Continuity stems from the continuity on each argument as in Propositions 6 and 7. Algebraicity also, remembering that the compact elements of a

finite product are the products of compact elements and that the minimum of the product is the pair of bottom elements. □

We define *practical knowledge* after observing u and applying the cognitions leading to S as Val_u^S. By Theorem 1 knowledge is exactly that which has been learnt.

4.2 Skill: Anomaly Detection

In order to present the *monitoring skill*, it is adequate to introduce a few notations. Let $u_{\lim} \in D_0$ be the ideal that contains all possible observations that contain no anomaly. It is assumed that learning is based on an observation $u \sqsubseteq u_{\lim}$. Assuming the signature is fixed, it is possible to define a limit valuation.

Definition 3. the *limit valuation* is $\mathrm{Val}_{u_{\lim}}^{\mathrm{Repr}(\mathrm{T}(\Sigma))}$.

Example 6. In our running example, the table of runs would contain all the possible runs, past, present, and future, and the minimum time would be the minimum amount of a time a human can run on 100m.

Since learning is continuous the valuation learnt after an observation u approximates the limit valuation and thus is a provisory estimate. *Monitoring* is the act of checking that an observed world u, which is not necessary a possible one, is within the limits learned.

Definition 4. A *ground literal* is an expression $t \sqsubseteq d$ where $t \in \mathrm{T}(\Sigma)$ is a representation term, and $d \in \mathrm{coDom}(t)$.

Monitoring is a classification problem with two classes. Remembering that $1 \sqsubseteq 0$ an observation contains a new behaviour whenever one of the possible representations is outside the bounds reached during learning.

Definition 5. (Monitor) Let Σ be a signature. Then for $r \in \Pi_{t \in \mathrm{Repr}(\mathrm{T}(\Sigma))} D(t)$, the r-monitor is the mapping:

$$\begin{array}{rccl} \mathcal{M}_r : & D_0 & \to & \mathbb{B} \\ & u & \mapsto & \sqcup_{t \in \mathrm{Repr}(\mathrm{T}(\Sigma))} \mathrm{Val}_u(t) \sqsubseteq r_t \end{array}$$

In the following lemma algebraicity is trivial as the two elements of $\mathbb{B}$ are compact, and continuity follows from the continuity of the valuation function (Lemma 1).

Lemma 2. *Let Σ be a signature, and d be in $\mathrm{coDom}(t)$ for a representation term t. Then the function $\mathcal{M}^{t,d} : u \mapsto \mathrm{Val}_u(t) \sqsubseteq d$ is algebraic. If furthermore d is compact then the function: $\mathcal{M}_s^{t,d} : u \mapsto \mathrm{Val}_u(t) \sqsubset d$ is continuous.*

Proof. Only the last statement is not trivial. Assume $u = \bigsqcup_{v \in V} v$ where V contains only compact elements. We have to prove that $\mathrm{Val}_u(t) \sqsubset d = \bigsqcup_{v \in V} \mathrm{Val}_v(t) \sqsubset d$. By

monotony of the valuation and $1 \sqsubset 0$ this is true if $\mathrm{Val}_u(t) \sqsubset d = 1$. Let us now assume this is not the case, and thus that $\mathrm{Val}_u(t) \sqsubset d = 0$ or equivalently that $d \sqsubseteq \mathrm{Val}_u(t)$.

The function $w \mapsto \mathrm{Val}_w(t)$ is continuous by Lemma 1 and thus $\mathrm{Val}_u(t) = \bigsqcup_{v \in \downarrow u} \mathrm{Val}_v(t)$. Since this function is algebraic all the $\mathrm{Val}_v(t)$ values are compact. By definition of compactness, and since algebraic lattices are bounded-complete, if $d \sqsubseteq \mathrm{Val}_u(t) = \bigsqcup_{v \in V} \mathrm{Val}_v(t)$ there exists $v \in V$ such that $d \sqsubseteq \mathrm{Val}_v(t)$ and thus such that $\mathrm{Val}_v(t) \sqsubset d = 0$. □

Proposition 8. *Let $r \in \Pi_{t \in \mathrm{Repr(T(\Sigma))}} \mathrm{coDom}(t)$, and $\mathcal{M}_r$ be the r-monitor. Then the mapping $u \in D_0 \mapsto \mathcal{M}_r(u) \in \mathbb{B}$ is algebraic.*

Note that in Proposition 8 the r-monitor is fixed. We leave to the reader that reusing the notations of Prop. 8 the function: $\psi_u : r \mapsto \mathcal{M}_r(u)$ is not continuous. This fact is exploited in the next section.

Example 7. Assuming that after an observation u a minimum time for sprinting on 100m by human is computed to be 9s58, and that an event of a run in 9s is observed. The monitor function maps that event to 0, and it is thus considered as anormal, to be treated depending on the context.

As this paper generalises the setting in [34] we refer to that paper for assessing the effectiveness in practice of this approach for anomaly detection.

4.3 Improving skills

Recall any non-anomalous observation u should be smaller than u_{lim}. Accordingly a world $u \not\sqsubseteq u_{\mathrm{lim}}$ is an *anomaly*. An anomaly u is Σ-*detectable* if $\mathcal{M}_{\mathrm{Val}_{u_{\mathrm{lim}}}^{\mathrm{Repr(T(\Sigma))}}}(u) = 0$. It is a r-false negative if $\mathcal{M}_r(u) = 1$. Finally we say that $u \sqsubseteq u_{\mathrm{lim}}$ is a r-false positive if $\mathcal{M}_r(u) = 0$, *i.e.*, it lies in the gap between the learnt valuation and the limit valuation.

Angluin-like [8] is synonymous with the possibility to converge to the right solution whenever false positive and false negatives examples to a proposed solution are given. The following theorem indicates that an Oracle can guide the system in the convergence to the limit valuation. In the first case it is only stated that a term exists. However we note that in settings in which [35] can be employed this computation automatable.

Theorem 2. *(Angluin-like Machine Learning) Let Σ be a signature, and $(u, S) \in \mathcal{D}(\Sigma)$ be a DXS. Let $r = \mathrm{Val}_u^{\mathrm{Repr}(S)}$ be the result of the learning phase, and $v \in D_0$ be a world.*

1. *if v is a detectable anomaly and a r-false negative, there exists S' such that $S \subseteq S'$ and $\mathcal{M}_{\mathrm{Val}_u^{\mathrm{Repr}(S')}}(v) = 0$;*
2. *if v is a normal behavior and a r-false positive, there exists u' such that $u \sqsubseteq u'$ and $\mathcal{M}_{\mathrm{Val}_{u'}^{\mathrm{Repr}(S)}}(v) = 1$.*

Proof. By the algebraicity and continuity of the monitor fonction. We prove the first case, the second one can either be proved similarly or by taking $u' = u + v$.

Since v is Σ-detectable, $\mathcal{M}_{\mathrm{Val}_{u_{\lim}}^{\mathrm{Repr}(\mathrm{T}(\Sigma))}}(v) = 0$. Since it is a r-false negative, $\mathcal{M}_{\mathrm{Val}_{u}^{\mathrm{Repr}(S)}}(v) = 1$. Since the function Val is algebraic continuous by Theorem 1, by considering an increasing chain of compacts (u', S') above (u, S) and whose supremum is $(u_{\lim}, \mathrm{T}(\Sigma))$, there exists a compact DXS (u', S') such that $u \sqsubseteq u'$, $S \subseteq S'$, and $\mathcal{M}_{\mathrm{Val}_{u'}^{\mathrm{Repr}(S')}}(v) = 0$. Since we take the supremum of the values on each term, there exists a term $t \in S'$ such that $\mathrm{Val}_v(t) \not\sqsubseteq \mathrm{Val}_{u'}(t)$. Let us consider the possibilities for t.

Since the valuation is increasing on each term, for all $t \in S$ we have $\mathrm{Val}_u(t) \sqsubseteq \mathrm{Val}_{u'}(t)$, and thus $\mathrm{Val}_v(t) \not\sqsubseteq \mathrm{Val}_{u'}(t)$ implies $\mathrm{Val}_v(t) \not\sqsubseteq \mathrm{Val}_u(t)$. Since v is a false negative, we have for all $t \in S$ that $\mathrm{Val}_v(t) \sqsubseteq \mathrm{Val}_u(t)$. Thus there exists $t \in S' \setminus S$ such that $\mathrm{Val}_v(t) \not\sqsubseteq \mathrm{Val}_{u'}(t)$, and thus $\mathrm{Val}_v(t) \not\sqsubseteq \mathrm{Val}_u(t)$. Thus we have $\mathcal{M}_{\mathrm{Val}_{u}^{\mathrm{Repr}(S')}}(v) = 0$. □

Conclusion on learning and monitoring. The continuity of the learning process established in Theorem 1 implies the eventual convergence of the result of our learning algorithm towards an ideal description of the system. Theorem 2 precises how to eliminate false positives–by learning more traces–,and false negatives–by adding new terms to the DXS.

5 Conceptual Knowledge in the Finite

Practical knowledge is useful for anomaly detection but does not address the discursive part of logic as presented in [5], namely that the mental state built through cognitions is the basis of discurse. While objects of perceptions and language are shared reality, the practical knowledge is subjective. The *unity of self* is the constraint that in particular names and relations in the shared language should be given a semantics consistent with the shared reality of the objects of perception. The concepts underlying this semanticisation are defined in this section as *predicates of experience* and *domain of experience*. The *conceptual knowledge* is the first-order model defined by these predicates and this domain.

Additional notations. Let $\mathcal{X}$ be a set of sorted variables denoted $x, y, \ldots$, and decorations thereof. A term t is a term in the signature $\mathrm{T}(\Sigma \cup \mathcal{X})$. The set of variables occuring in t is $\mathrm{Sub}(t) \cap \mathcal{X}$. A term t is a *pure representation term* if all functions in the definition of t are representation functions.

Outline. A *state* is a couple (u, S) where u is an element of D_0 and S is a DXS. Concepts are introduced through new cognitions that enrich a *cognition state*. These cognition states are states extended with a subset $O \subseteq S$ of ground object terms an observer *is aware of* when reasoning and a set R of pure representation terms through which these objects are classified. A first-order model is built by concept analysis on the object terms as objects and pure representation terms as attribute.

5.1 Cognitive states

A literal of the form $t \sqsubseteq d$ where t is a pure representation term is called a *pure literal*.

Definition 6. (Predicate) A *predicate* $\varphi = \bigwedge_{t \in T_\varphi} t \sqsubseteq d_t$ is a conjunction of pure literals. Its *arity* is $|\cup_{t \in T_\varphi} \mathrm{Var}(t)|$.

The truth of a pure literal l in relation with an observation u is defined through one or several substitutions θ such that $l\theta$ is ground and satisfied by u.

Definition 7. (Grounding) Let O be a set of object terms and t be a pure representation term. A substitution θ *O-grounds* t if:

$$\begin{cases} \mathrm{coDom}(\theta) \subseteq O \\ t\theta \text{ is ground} \end{cases}$$

We denote $\mathrm{Gr}_O(t)$ the set of substitutions that O-grounds the term t.

Given a set of object terms O we denote $\mathrm{Subst}(O)$ the set of substitutions whose codomain is included in O. As a matter of convenience this notion is extended to literals and we denote $\mathrm{Gr}_O(t \sqsubseteq d)$ the set $\mathrm{Gr}_O(t)$ when t is a pure representation term.

Definition 8. (Support of a literal) Let O be a set of object terms and $u \in D_0$. A substitution θ *(u, O)-supports a pure literal* $r \sqsubseteq d$ if $\theta \in \mathrm{Gr}_O(r \sqsubseteq d)$ and $\mathrm{Val}_u(r\theta) \sqsubseteq d$. We denote this fact with $\theta \models_{(u,O)} l$, and denote $\mathrm{Supp}_u^O(l)$ the subset of $\mathrm{Gr}_O(l)$ of substitutions that (u, O)-supports the literal l.

Definition 9. (Cognitive State) A *cognitive state* is a tuple $K = (u, S, O, R)$ such that:

- $u \in D_0$ and S is a DXS;
- $O \subseteq S$ contains only object terms and for all $o \in O$ we have $[o]_u^S \neq \varepsilon$;
- R is a set of pure representation terms;

Given two cognitive states $K = (u, S, O, R)$ and $K' = (u', S', O', R')$ we say that K' is an extension K, and denote it $K \sqsubseteq K'$, if $u \sqsubseteq u'$, $S \subseteq S'$, $O \subseteq O'$, and $R \subseteq R'$.

Given a family of cognitive states $(K_f)_{f\in F}$ the supremum of the family is denoted $\bigsqcup_{f\in F} K_f$ and is the cognitive state $K = (u, S, O, R)$ where:

$$\begin{cases} u &= \cup_{f\in F} u_f \\ S &= \cup_{f\in F} S_f \\ R &= \cup_{f\in F} R_f \\ O &= \cup_{f\in F} O_f \end{cases}$$

By altering the construction in Sec. Section 3.2 it is clear that the set of cognitive states is a domain, *i.e.*, an algebraic lattice. The set of cognition functions is extended with cognitions Add_t^O adding the term t to O (with $\mathrm{Add}_t \sqsubseteq \mathrm{Add}_t^O$ to ensure that only terms in S can be added, and Add_t^R to add pure representation terms to R. As a consequence cognitive state $K = (u, S, O, R)$ is *compact* whenever u and S are compact, and O and R are finite.

5.2 Relational signature of a cognitive state

Given a cognitive state $K = (u, S, O, R)$ and $t \in R$ let $\mathrm{Gr}_K(t) \subseteq \mathrm{Gr}_O(t)$ be the set of substitutions θ such that $t\theta \in S$. The set of substitutions *defined* in a cognitive

state K is denoted $\mathrm{Subst}(K)$ and is the set $\cup_{t\in R}\mathrm{Gr}_K(t)$. Given a substitution $\theta \in \mathrm{Subst}(K)$, we let $\mathrm{Obs}_K(\theta)$ be the set of *observations* that can be made on the objects in the image of θ:

$$\mathrm{Obs}_K(\theta) = \{t \in R \,|\, \theta \in \mathrm{Gr}_K(t)\}$$

We extend this notation to sets of substitutions with:

$$\mathrm{Obs}_K(\Theta) = \cap_{\theta\in\Theta}\mathrm{Obs}_K(\theta)$$

Example 8. Let $K = (u, S, O, R)$ be a cognitive state, $f : T \to T_D$ be a representation function, and $\mathrm{eq} : T_D \times T_D \to \{0, 1\}$ be an equality function, *i.e.*, such that $\mathrm{eq}(u, v) = 1$ if and only if $u = v$. The representation term $\mathrm{eq}(f(x), f(y))$ is a binary predicate. Let Θ be the set of substitutions $\theta = \{x \mapsto t_1, y \mapsto t_2\}$ with $t_1, t_2 \in O$ such that $\mathrm{eq}(f(t_1), f(t_2)) = 1$. By definition we have $\mathrm{eq}(f(x), f(y)) \in \mathrm{Obs}_K(\Theta)$. Knowing that the predicate must be reflexive, symmetric, and transitive can help in the computation of Θ, and symmetrically the knowledge of a maximal set of substitutions Θ leads to the learning from experience that the predicate is reflexive, symmetric, and transitive. These cognitive aspects are out of the scope of this paper, but the construction of rules from table contents given in [24] can be employed.

Since the set $\mathrm{Obs}_K(\Theta)$ represents the common qualities of the objects related by the substitutions in Θ *predicates of experience* as a subset A of particular aspects of these common qualities.

Definition 10. (Predicate of Experience) Let K be a cognitive state, and $\Theta \subseteq \mathrm{Subst}(K)$. The *predicate of experience defined by* Θ *and* $A \subseteq \mathrm{Obs}_K(\Theta)$ *in* K is denoted $\mathrm{Obj}^A_K(\Theta)$ and is the formula:

$$\mathrm{Obj}^A_K(\Theta) = \begin{cases} 1 & \text{If } A = \emptyset \\ \bigwedge_{t\in A} t \sqsubseteq \bigsqcup_{\theta\in\Theta}\mathrm{Val}_u(t\theta) & \\ & \text{Otherwise} \end{cases}$$

Its *arity* is the cardinal of $\cup_{t\in A}\mathrm{Var}(t)$. The set of predicates of experience of a set of substitutions Θ is denoted $\mathrm{Obj}_K(\Theta)$.

We denote $\mathrm{Obj}^E(K)$ the set of predicates of experience $\mathrm{Obj}^A_K(\Theta)$ for $\Theta \subseteq \mathrm{Subst}(K)$ and $A \subseteq \mathrm{Obs}_K(\Theta)$. Gr_K is extended to literals, and to predicates with $\mathrm{Gr}_K(\bigwedge_{t\in T} t \sqsubseteq d_t) = \cap_{t\in T}\mathrm{Gr}_K(t)$. We give the usual semantics to the conjunction by defining the *support of* $\varphi = \bigwedge_{t\in T} t \sqsubseteq d_t$ *in* K as $\cap_{t\in T}\mathrm{Supp}^O_u(t \sqsubseteq d_t)$, and denote it $\mathrm{Supp}^O_u(\varphi)$. By construction $\Theta \subseteq \mathrm{Supp}^O_u(\varphi)$ for $\varphi \in \mathrm{Obj}_K(\Theta)$: a predicate defined by the examples in Θ is satisfied by these examples. It is trivial that if $\theta \in \mathrm{Supp}^O_u(\mathrm{Obj}^{\mathrm{Obs}_K(\Theta)}_K(\Theta))$ then $\theta \in \mathrm{Supp}^O_u(\varphi)$ for all $\varphi \in \mathrm{Obj}_K(\Theta)$. That case is denoted $\theta \in \mathrm{Supp}^O_u(\mathrm{Obj}_K(\Theta))$, and extended to sets fo substitutions Θ'.

That a set of substitutions may entail the object formulas of another set of substitutions yields a pre-order on these sets. In turns the pre-order yields an equivalence relation between sets of substitutions.

Definition 11. (Specialisation) Let K be a cognitive state, and $\Theta, \Theta' \subseteq \mathrm{Subst}(K)$. We say that Θ' *specialises* Θ, and denote $\Theta \preceq_K \Theta'$, if $\Theta' \subseteq \mathrm{Supp}^O_u(\mathrm{Obj}_K(\Theta))$.

The sets $\Theta, \Theta' \subseteq \mathrm{Subst}(K)$ are K*-equivalent*, and we denote $\Theta \equiv_K \Theta'$, if $\Theta \preceq_K \Theta'$ and $\Theta' \preceq_K \Theta$.

The equivalence classes for $\equiv_K$ are (pre-order) isomorphic with the predicates of experience on K.

Lemma 3. *Let $K = (u, S, O, R)$ be a cognitive state, and $\Theta, \Theta' \subseteq \mathrm{Subst}(K)$ be such that $\Theta \equiv_K \Theta'$. Then $\mathrm{Obj}_K(\Theta) = \mathrm{Obj}_K(\Theta')$.*

Proof. Let $\Theta, \Theta' \subseteq \mathrm{Subst}(K)$ be such that $\Theta \preceq_K \Theta'$. It suffices to proves $\mathrm{Obj}_K^{\mathrm{Obs}_K(\Theta)}(\Theta) = \mathrm{Obj}_K^{\mathrm{Obs}_K(\Theta')}(\Theta')$.

By definition $\Theta \preceq_K \Theta'$ implies $\mathrm{Obs}_K(\Theta') \subseteq \mathrm{Obs}_K(\Theta)$, and thus by double inclusion $\Theta \preceq_K \Theta'$ implies $\mathrm{Obs}_K(\Theta) = \mathrm{Obs}_K(\Theta')$.

By definition of $\preceq_K$ for each $\theta' \in \Theta'$ we have $u \models \mathrm{Obj}_K(\Theta)\theta'$. For each $r \in \mathrm{Obs}_K(\Theta)$, $u \models \mathrm{Obj}_K(\Theta)\theta'$ implies $\mathrm{Val}_u(r\theta') \sqsubset \bigsqcup_{\theta \in \Theta} \mathrm{Val}_u(r\theta)$. Inversing the roles of θ, θ' we also get that for all $r \in \mathrm{Obs}_K(\Theta')$, and all $\theta \in \Theta$ we have $\mathrm{Val}_u(r\theta) \sqsubset \bigsqcup_{\theta' \in \Theta'} \mathrm{Val}_u(r\theta')$. Taken together these inequations yield for all $r \in \mathrm{Obs}_K(\Theta)$:

$$\bigsqcup_{\theta \in \Theta} \mathrm{Val}_u(r\theta) = \bigsqcup_{\theta' \in \Theta'} \mathrm{Val}_u(r\theta')$$

Given the two preceding paragraphs and the definition of Obj_K we have: $\mathrm{Obj}_K(\Theta) = \mathrm{Obj}_K(\Theta')$.

Conversely, $\mathrm{Obj}_K(\Theta) = \mathrm{Obj}_K(\Theta')$ implies, together with the fact that by construction $u \models \mathrm{Obj}_K(\Theta)\theta'$ for all $\theta' \in \Theta'$, and symmetrically when reversing the roles of Θ and Θ', that $\Theta \equiv_K \Theta'$. □

This tight coupling is employed to transfer the predicates of experience between cognitive states through the sets of substitutions and representations that generate them.

5.3 Domain of a cognitive state

The construction of a domain from a cognitive state $K = (u, S, O, R)$ proceeds by considering the elements of O modulo an equivalence generated by $\mathrm{Obj}^E(K)$. This approach has two obstacles:

- A first modeling problem is whether it suffices to consider all comparisons between terms in O in the image of the possible substitutions, or if all the replacement of one term by another in O have to be considered. Both choices lead to an equivalence relation on O, the latter making it a congruence on terms. Lemma 5 provides the proof of termination of the replacements though to ease notations only the changes in the codomain of the substitutions are considered outside of the corresponding paragraph, with hints for the non-obvious adaptations;
- A second problem is that pairwise comparison of terms does not yield a transitive relation as some terms may be missing in S. Instead of considering all intermediate cognition states as described by Kant, we introduce the notion of *stable state* to indicate that all the needed replacements have been performed.

Example 9. Importance of transitivity for proper reasoning. It can be determined that a witch weights the same as a duck by considering successively that people burn witches and wood, that wood and ducks float on water, and that the weight

of an object determines whether it floats. The cognitive state on which this reasoning is sensible lacks transitivity as it is not determined whether people burn ducks nor whether witches float. It is thus not stable, and additional experiences (construction of new terms) are needed to make it stable. Out of the scope of this paper, it would be also sensible to determine whether floating on water is a good characterisation of witches as a special subset of humans.

Considering closure under replacements. The following lemma allows for the transfer of a replacement on an instantiated term to a replacement on the substitution, and is used without references when considering ground instances of pure representation terms.

Lemma 4. *Let t be an object term, r be a pure representation, and θ be a substitution such that $r\theta$ is ground. Then if $(r\theta)_{|p} = t$ and $p \in \mathrm{Pos}(r)$ then $r_{|p}$ is a variable.*

Proof. Since t is an object term the symbol at position ϵ in t denotes an object morphism. Since r is a pure representation term, this symbol cannot occur in r. Thus t can occur in $r\theta$ only at a position of a variable or below. □

Definition 12. A set of ground terms O is *replacement saturated* if for all $t, t_1, t_2 \in O$ and position $p \in \mathrm{Pos}(t)$, we have either $t_{|p} = t_1$ implies $t[p \leftarrow t_2] \in O$ or $t_{|p} = t_2$ implies $t[p \leftarrow t_1] \in O$.

The replacement saturation of a set of terms can be computed effectively and is finite.

Lemma 5. *Let $K = (u, S, O, R)$ be a compact cognitive state. Then there exists a finite and unique minimal set of ground term O' such that $O \subseteq O'$ and O' is replacement saturated.*

Proof. Let $<$ be a simplification ordering [22] on terms. We consider the ground term rewriting system $T = \{t \to t' \,|\, t, t' \in O \text{ and } t' < t\}$. The Knuth-Bendix completion of T always succeed (Corollary 6.2 in [10]) and yields a finite equivalent term rewriting sytem T'. Let O' be the set of terms that contains O and such that $t \in O'$ and $t \to^{T'} t'$ implies $t' \in O'$. Since T' is always terminating this set if finite, and it is replacement saturated by definition. □

Proper construction of the domain. From now on we only consider the replacement in the codomain of substitutions of one term with another term. This replacement may be partial. Given a substitution θ of support X, for $Y \subseteq X$ and t, t' terms we denote $\theta[t \leftarrow t']_Y$ the substitution:

$$\theta[t \leftarrow t']_Y(x) = \begin{cases} \theta(x) & \text{if } x \notin Y \\ t' & \text{if } x \in Y \text{ and } \theta(x) = t \end{cases}$$

The following definition can be adapted by furthermore assuming O is replacement saturated and considering the orbits for the symmetric closure of the $t \to t'$ rewriting rule: the terms t and t' are equivalent if and only if the valuation is constant over the orbit of each representation term.

Definition 13. (Pre-order and equivalence on O) Let $K = (u, S, O, R)$ be a compact cognitive state. Given $t, t' \in O$ we write $t \preceq^o_K t'$ if for all $\theta \in \text{Subst}(K)$, for all $X \subseteq \text{Dom}(\theta)$, and for all $r \in R$ we have $\text{Val}^S_u(r\theta[t \leftarrow t']_X) \sqsubseteq \text{Val}^S_u(r\theta)$.

We denote $t \equiv^o_K t'$ the fact that $t \preceq^o_K t'$ and $t' \preceq^o_K t$.

This closure by partial replacement may seem too stringent but is technically necessary to prove transitivity in the following lemma.

Lemma 6. *$t \preceq^o_K t'$ is pre-order on O.*

Proof. It is trivially reflexive. Assume $t, t', t'' \in O$ be such that $t \preceq^o_K t'$ and $t' \preceq^o_K t''$.

Let $\theta \in \text{Subst}(K)$, $X \subseteq \text{coDom}(\theta)$, and $r \in R$. We remark that $\theta[t \leftarrow t'']_X = \theta[t \leftarrow t'']_{X \cap \theta^{-1}(t)}$. It thus suffices to prove $\text{Val}^S_u(r\theta[t \leftarrow t'']_{X\cap\theta^{-1}(t)}) \sqsubseteq \text{Val}^S_u(r\theta)$.

By composition we have $r\theta[t \leftarrow t'']_{X\cap\theta^{-1}(t)} = r\theta[t \leftarrow t']_{X\cap\theta^{-1}(t)}[t' \leftarrow t'']_{X\cap\theta^{-1}(t)}$ and conclude with the hypothesis and the transitivity of $\sqsubseteq$. □

Thus by Lemma 6 $\equiv^o_K$ is an equivalence relation on O. The behaviour of this equivalence is problematic when considering the addition of representation terms in S. Assume $O = \{t_1, t_2\}$, $[t_1]^S_u = [t_2]^S_u$ and $R = \{f(x)\}$. If S contains only one of $f(t_1), f(t_2)$ we necessarily have $t_1 \not\equiv^o_K t_2$ but if it contains either none or both of them we have $t_1 \equiv^o_K t_2$. Starting from an empty set O and adding successively t_1 and t_2 makes the equivalence classes not monotonic. Accordingly *stable* cognitive states are those states such that S contains enough terms to properly evaluate all replacements of one term in O with another when instantiating a term in R.

Definition 14. (Stable cognitive state) A cognitive state $K = (u, S, O, R)$ is *stable* if for all $r \in R$ we have $\text{Subst}_K(r) = \text{Subst}_O(r)$.

The essence of the following proposition is that the equivalence class $\equiv^o_K$ on O defines a finer equivalence than $\equiv_K$ on $\mathcal{P}(\text{Subst}(K))$ (the first part), but that it is the coarsest equivalence on O that induces an equivalence finer than $\equiv_K$ on $\mathcal{P}(\text{Subst}(K))$ (the second part).

Proposition 9. *Let K be a stable cognitive state, and $t, t' \in O$. If $t \equiv^o_K t'$ then for all $\Theta \subseteq \text{Subst}(K)$ and $X \subseteq \cap_{\theta\in\Theta} \text{Dom}(\theta)$ we have*

$$\text{Obj}_K(\Theta) = \text{Obj}_K(\{\theta[t \leftarrow t']_X \mid \theta \in \Theta\}$$

Conversely if for all $\Theta \subseteq \text{Subst}(K)$ and $X \subseteq \text{Dom}(\Theta)$ we have

$$\text{Obj}_K(\Theta) = \text{Obj}_K(\{\theta[t \leftarrow t']_X \mid \theta \in \Theta, X \subseteq \text{Dom}(\theta)\})$$

then $t \equiv^o_K t'$.

Proof. If $t \equiv^o_K t'$ for all $\Theta \subseteq \text{Subst}(K)$ and $X \subseteq \cap_{\theta\in\Theta} \text{Dom}(\theta)$ we have for all $\theta \in \Theta$ that $\theta \equiv_K \theta[t \leftarrow t']_X$ by definition of $\equiv^o_K$, and thus $\Theta \equiv_K \Theta[t \leftarrow t']_X$ by definition of $\equiv_K$.

Conversely assume that for all $\Theta \in \mathrm{Subst}(K)$ and all $X \subseteq \cap_{\theta\in\Theta} \mathrm{Dom}(\theta)$ we have

$$\mathrm{Obj}_K(\Theta) = \mathrm{Obj}_K(\{\theta[t \leftarrow t']_X \,|\, \theta \in \Theta\})$$

By Lemma 3 $\mathrm{Obj}_K(\Theta) = \mathrm{Obj}_K(\{\theta\delta_{t,t'} \,|\, \theta \in \Theta\} \cap \mathrm{Subst}(K))$ is equivalent to $\Theta \equiv_K \{\theta[t \leftarrow t']_X \,|\, \theta \in \Theta\}$.

In particular for all $\theta \in \Theta$ and for all $r \in R$ such that $\theta \in \mathrm{Gr}_K(r)$ that $\mathrm{Val}^S_{r\theta[t\leftarrow t']_X}(\sqsubseteq)\,\mathrm{Val}^S_u(r\theta)$, and thus again by Definition 13 that $t \equiv^o_K t'$. □

Next lemma is a direct consequence of Definition 13 that is highlighted for further reference.

Lemma 7. *Let $K = (u, S, O, R)$ be a compact stable state and $t_1, t_2 \in O$ such that $t_1 \equiv^o_K t_2$. Then for all $\theta \in \mathrm{Subst}(K)$, for all $X \subseteq \mathrm{Dom}(\theta)$, and for all $r \in \mathrm{Obs}_K(\theta)$ we have $r\theta = r\theta[t_1 \leftarrow t_2]_X$.*

Thus the truth value of predicates of experience can be defined on the equivalence classes of $\equiv^o_K$. This approach fails to take into account that there exists objects not in O. Accordingly given the domains $D_0, \ldots, D_n$ let *Others* $= \{\mathit{other}_D\}_{D\in\{D_0,\ldots,D_n\}}$ be a set of constants each interpreted in its labeling domain as the bottom of that domain. Even if different terms in an equivalence class may have different interpretation, Lemma 7 implies when considering predicates of experience, the choice of the representative in indifferent. Thus these two sets of constants together define a domain on which predicates of experience can be evaluated.

Definition 15. (Domain of a cognitive state) Let $K = (u, S, O, R)$ be a stable cognitive state. The *domain of* K is the set $O/\equiv^o_K \cup \mathit{Others}$ and is denoted $\mathrm{Dom}(K)$.

Valuation on $\mathrm{Dom}(K)$**.** We denote the valuation on $\mathrm{Dom}(K)$ with Val^K and define:

$$\mathrm{Val}^K(a) = \begin{cases} \bot_{D_i} & \text{If } a \in \mathit{Others} \text{ of sort } \tau_i \\ \mathrm{Val}_u(t) & \text{for any } t \in a \text{ otherwise} \end{cases}$$

5.4 Conclusion

By Lemma 7 the valuation of an instantiation of a term $r \in R$ never depends on the term chosen in the equivalence class. This allows the extension of substitutions to the domain of a cognitive state.

Definition 16. (Model a Compact Cognitive State) Let $K = (u, S, O, R)$ be a stable compact cognitive state. The *model of* K is denoted $\mathrm{Mod}(K)$ and is the tuple $(\mathrm{Dom}(K), \mathrm{Obj}^E(K), \models_K)$. Given $P \in \mathrm{Obj}^E(K)$ of arity n we denote $\models_K P(a_1, \ldots, a_n)$ iff $\theta = \{x_i \mapsto a_i\}_{1\le i\le n} \in \mathrm{Supp}^O_u(R)$.

A preliminary of *conceptual knowledge* for a stable compact cognitive state K is $\mathrm{Mod}(K)$.

6 Conceptual Knowledge

To bridge the gap with [30] it suffices to construct an inverse system for conceptual knowledge.

6.1 Analysis of Predicates of Experience

Let $K = (u, S, O, R)$ be a stable compact cognitive state. Let $P(x_1, \ldots, x_n)$ be a predicate of experience in $\mathrm{Obj}^E(K)$. there exists an equivalence class $\Theta_P \subseteq \mathrm{Subst}(K)$ and $A \subseteq \mathrm{Obs}_K(\Theta_P)$ such that:

$$R(x_1, \ldots, x_n) = \bigwedge_{r \in A} r \sqsubseteq \bigsqcup_{\theta \in \Theta_P} \mathrm{Val}_u(r\theta)$$

with $\bigcup_{r \in A} \mathrm{Var}(r) = \{x_1, \ldots, x_n\}$.

First assume $\Theta'_P \subseteq \Theta$. Then for all r $\bigsqcup_{\theta \in \Theta'_P} \mathrm{Val}_u(r\theta) \sqsubseteq \bigsqcup_{\theta \in \Theta_P} \mathrm{Val}_u(r\theta)$, and thus if

$$\bigwedge_{r \in A} r \sqsubseteq \bigsqcup_{\theta \in \Theta'_P} \mathrm{Val}_u(r\theta)$$

is satisfied by a substitution σ then

$$\bigwedge_{r \in A} r \sqsubseteq \bigsqcup_{\theta \in \Theta_P} \mathrm{Val}_u(r\theta)$$

is also satisfied by σ. However we also note that by construction, if this substitution σ is in Θ'_P, it satisfies both formulas.

Second, assume now that $R' \subseteq R$, and let $A' = A \cap R'$. Since the valuation of each formula does not change, it is tautological that if a substitution σ satisfies:

$$\bigwedge_{r \in A} r \sqsubseteq \bigsqcup_{\theta \in \Theta_P} \mathrm{Val}_u(r\theta)$$

it also satisfies:

$$\bigwedge_{r \in A'} r \sqsubseteq \bigsqcup_{\theta \in \Theta_P} \mathrm{Val}_u(r\theta)$$

Finally assume that $K' = (u', S', O', R')$ is also a stable compact cognitive state with $u = u'$ and $K' \sqsubseteq K$, *i.e.*, $O' \subseteq O$ and $R' \subseteq R$.

By definition $O' \subseteq O$ implies $\mathrm{Subst}(K') \subseteq \mathrm{Subst}(K)$. Let Θ_P be an equivalence class in $\mathrm{Subst}(K)$, let $A \subseteq \mathrm{Obs}_K(\Theta_P)$, and let:

$$P(x_1, \ldots, x_n) = \bigwedge_{r \in A} r \sqsubseteq \bigsqcup_{\theta \in \Theta_P} \mathrm{Val}_u(r\theta)$$

Define:

$$\left\{ \begin{array}{rl} \Theta_{P'} = & \left\{ \theta_{|\mathrm{Var}(P')} \,|\, \theta \in \Theta_P \right\} \cap \mathrm{Subst}(K') \\ A' = & A \cap \mathrm{Obs}'_K(\Theta_{P'}) \\ P'(x_1, \ldots, x_m) = & \bigwedge_{r \in A'} r \sqsubseteq \bigsqcup_{\theta \in \Theta_{P'}} \mathrm{Val}_u(r\theta) \end{array} \right.$$

Two cases are possible:

- If $\Theta_{P'}$ is empty, $\mathrm{Obs}_{K'}(\emptyset) = \emptyset$, and thus $A' = \emptyset$ and $m = 0$. Thus the conjunction is always true, *i.e.*, $P' = 1$. Any substitution satisfying $P(x_1, \ldots, x_n)$ also satisfies 1;

- Otherwise from the above discussion we have that any substitution $\theta \in \Theta_{P'}$ satisfies both $P(x_1, \ldots, x_n)$ and $P'(x_1, \ldots, x_m)$. Thus again the image of any substitution that satisfies $P(x_1, \ldots, x_n)$ satisfies $P'(x_1, \ldots, x_m)$

By construction $P'(x_1, \ldots, x_m)$ is a predicate of experience of K'. This reasoning applies on all predicate of experience of K, and given that $R' \subseteq R$ and $\text{Subst}(K') \subseteq \text{Subst}(K)$ one easily obtains $\text{Obj}^E(K') \subseteq \text{Obj}^E(K)$. This case is generalised with functions between any stable compact cognitive states K', K such that $K' \sqsubseteq K$.

6.2 Projections of the Inverse System

Definition 17. (Predication Projection) Let $K = (u, S, 0, R), K' = (u, S', O', R')$ be two stable compact cognitive states with $K' \sqsubseteq K$. The *predicate projection* of K into K' is denoted $\psi_{K',K}$ and is the function:

$$\begin{array}{rccc} \psi_{K',K}: & \text{Obj}^E(K) & \rightarrow & \text{Obj}^E(K') \\ & \text{Obj}^A_K(\Theta) & \mapsto & \text{Obj}^{A'}_K(\Theta') \end{array}$$

with:

$$\left\{ \begin{array}{lcl} A' & = & A \cap R' \\ \Theta' & = & \left\{\theta_{|\text{Var}(A')} \,|\, \theta \in \Theta_P\right\} \cap \text{Subst}(K') \end{array} \right.$$

The two first points of next lemma summarizes the discussion of the preceding section. The last point is by construction.

Lemma 8. *Let $K = (u, S, 0, R), K' = (u, S', O', R'), K'' = (u, S'', O'', R'')$ be stable compact cognitive states with $K'' \sqsubseteq K' \sqsubseteq K$.*

1. *$\psi_{K',K}$ is surjective;*
2. *For all $P \in \text{Obj}^E(K)$ of arity n, for all $a_1, \ldots, a_m \in O'$ such that there exists $a_{m+1}, \ldots, a_n$ in O with $\models_K P(a_1, \ldots, a_n)$, we have $\models_{K'} \psi_{K',K}(P)(a_1, \ldots, a_m)$.*
3. *$\psi_{K'',K} = \psi_{K'',K'} \circ \psi_{K',K}$.*

Next lemma enables a similar construction on the domain.

Lemma 9. *Let $K = (u, S, 0, R), K' = (u, S', O', R')$ be two stable compact cognitive states with $K' \sqsubseteq K$, and $t_1, t_2 \in O'$ such that $t_1 \equiv^o_K t_2$. Then $t_1 \equiv^o_{K'} t_2$*

Proof. Trivial by Lemma 7 and $R' \subseteq R$. □

Thus if the intersection with O' of an equivalence class $a \in O/\equiv_K$ is not empty, it is a subset of a unique equivalence class $a' \in O'/\equiv_{K'}$. Let $h_{K',K}$ be provisionally defined as the mapping $a \in O/\equiv^o_K \mapsto a' \in O'/\equiv^o_{K'}$. Considering the image of the equivalence class containing each $t \in O' \subseteq O$ shows that this mapping is surjective. It is extended as follows to $\text{Dom}(K)$:

- If $a \in$ *Others* define $h_{K',K}(a) = a$;
- Otherwise if $a \cap O' = \emptyset$ and a is interpreted on the domain D define $h_{K',K}(a) = other_D$.

Definition 18. (Domain Projection) Let $K = (u, S, O, R), K' = (u, S', O', R')$ be two stable compact cognitive states with $K' \sqsubseteq K$. The *domain projection* of K into K' is denoted $h_{K',K}$ and is the function:

$$h_{K',K} : \quad \mathrm{Dom}(K) \quad \rightarrow \quad \mathrm{Dom}(K')$$
$$h_{K',K}(a) = \begin{cases} a & \text{If } o \in \textit{Others} \\ \textit{others}_d & \text{If } a \notin \textit{Others} \text{ and } a \cap O' = \emptyset \\ a' & \text{Otherwise} \end{cases}$$

with a' the equivalence class in O' containing $a \cap O'$.

Predicate and domain projections are related in the following lemma. The first point is by construction, the second by Lemma 8.

Lemma 10. *Let $K = (u, S, 0, R), K' = (u, S', O', R'), K'' = (u, S'', O'', R'')$ be stable compact cognitive states with $K'' \sqsubseteq K' \sqsubseteq K$. Then the predicate and domain projections of K into K' are surjective and*

- $h_{K'',K} = h_{K'',K'} \circ h_{K',K}$;
- *For all $a_1, \ldots, a_n \in \mathrm{Dom}(K)$ and $P(x_1, \ldots, x_n) \in \mathrm{Obj}^E(K)$ one has $\models_K P(a_1, \ldots, a_n)$ implies $\models_{K'} \psi_{K',K}(P)(h_{K',K}(a_1), \ldots, h_{K',K}(a_m))$.*

Lemma 8 and 10 directly prove Theorem 3. Let us first recall the definition of inverse systems of models.

Definition 19. (Inverse System of models) Let F be a directed poset, and $\{\mathcal{M}_f = (D_f, \mathcal{R}_f, \models_f)\}_{f \in F}$ be a family of first-order models. Then $((\mathcal{M}_f)_{f \in F}, (g_{f',f})_{f' \sqsubseteq f}, (\varphi_{f',f})_{f' \sqsubseteq f})$ is an *inverse system of models* if:

- *Coherence:* $g_{f',f} : D_f \rightarrow D_{f'}$ and $\varphi_{f,f'} : R_{f'} \rightarrow R_f$ are surjective mappings satisfying: $f \subseteq f' \subseteq f''$ implies $g_{f'',f} = g_{f'',f'} \circ g_{f',f}$ and $\varphi_{f'',f} = \varphi_{f'',f'} \circ \varphi_{f',f}$;
- *Model homomorphism:* for all $P \in \mathcal{R}_f$ there exists a subset $m_f \leq n$ such that $\models_f P(a_1, \ldots, a_n)$ implies $\models_{f'} \varphi_{f',f}(P)(g_{f',f}(a_1), \ldots, g_{f',f}(a_{m'_f}))$

Given a stable cognitive state $K = (u, S, O, R)$ let $K\downarrow$ be the set of stable compact cognitive states $K' = (u', S', O', R')$ such that $K' \sqsubseteq K$ and $u' = u$. We have:

Theorem 3. *Let $K = (u, S, O, R)$ be a stable cognitive state. Then*

$$((\mathrm{Mod}(K'))_{K' \in K\downarrow}, (h_{K'',K'})_{K'' \sqsubseteq K' \in K\downarrow}, (\psi_{K'',K'})_{K'' \sqsubseteq K' \in K\downarrow})$$

is an inverse system of models.

We deviate here from the treatment in [30] to prove that this inverse system of models has a non-empty inverse limit. Given $K'' \sqsubseteq K' \in K\downarrow$ define:

Definition 20. (Threads of an inverse system of models) Let F be a directed poset and

$$\mathcal{M} = ((\mathcal{M}_f = (D_f, \mathcal{R}_f, \models_f))_{f \in F}, (g_{f',f})_{f' \sqsubseteq f}, (\varphi_{f',f})_{f' \sqsubseteq f})$$

be an *inverse system of models*. Let $\Delta \subseteq \Pi_{f\in F} D_f$ be the set of all ξ such that $f \geq f'$ implies $g_{f',f}(\xi(f)) = \xi(f')$. Then Δ is the set of *threads* of $\mathcal{M}$.

Definition 21. (Predicates of an inverse system of models) Let F be a directed poset and

$$\mathcal{M} = ((\mathcal{M}_f = (D_f, \mathcal{R}_f, \models_f))_{f\in F}, (g_{f',f})_{f'\sqsubseteq f}, (\varphi_{f',f})_{f'\sqsubseteq f})$$

be an *inverse system of models*. Let $P \subseteq \Pi_{f\in F} R_f$ be the set of all ρ such that $f \geq f'$ implies $\varphi_{f',f}(\rho(f)) = \rho(f')$. Then P is the set of *predicates* of $\mathcal{M}$.

Definition 22. (Inverse Limit of an Inverse System) Let F be a directed set and

$$\mathcal{M} = ((\mathcal{M}_f = (D_f, \mathcal{R}_f, \models_f))_{f\in F}, (g_{f',f})_{f'\sqsubseteq f}, (\varphi_{f',f})_{f'\sqsubseteq f})$$

be an inverse system of models. Let Δ be the set of threads of $\mathcal{M}$ and P be its set of predicates. Let $M = (\Delta, P, \models_M)$ be a model where $\models_M \rho(\xi_1, \ldots, \xi_n)$ if for all $f \in F$ we have $\models_f \rho(f)(\xi_1(f), \ldots, \xi_{m_f}(f))$ with $m_f \leq n$ being the arity of $\rho(f)$.

Then M is the *inverse limit* of the inverse system $\mathcal{M}$.

In [30] the proof of the equivalent theorem relies on Tychonoff's Theorem and contingent considerations on threads. It is better viewed as a generic domain result. Indeed each mapping in $((D_f)_{f\in F}, (g_{f',f})_{f'\sqsubseteq f})$ and $((\mathcal{R}_f)_{f\in F}, (\varphi_{f',f})_{f'\sqsubseteq f})$ is a projection, and taking the lower adjoints of these defines two expanding systems with the indicated limit by Theorem 3.3.7 in [28].

Theorem 4. *Let F be a directed set and $\mathcal{M} = ((\mathcal{M}_f = (D_f, \mathcal{R}_f, \models_f))_{f\in F}, (g_{f',f})_{f'\sqsubseteq f}, (\varphi_{f',f})_{f'\sqsubseteq f})$ be an inverse system of models of inverse limit $M = (\Delta, P, \models_M)$. Then M is not empty.*

The inverse limit as defined in Definition 22 is consistent with our definition of models for stable compact cognitive states.

Lemma 11. *Let $K = (u, S, O, R)$ be a stable compact cognitive state. Then $Model(K)$ is isomorphic with the inverse limit of:*

$$(\mathrm{Mod}(K'))_{K'\in K\downarrow}, (h_{K'',K'})_{K''\sqsubseteq K'\in K\downarrow}, (\psi_{K'',K'})_{K''\sqsubseteq K'\in K\downarrow})$$

Proof. Since K is compact the projections define predicates and threads unambiguously: *e.g.,* for every thread xi in the domain of the inverse limit one has $\xi(K') = h_{K',K}(\xi(K))$. The model homomorphism property of inverse system entails that the interpretation of the limit in the inverse system is the same as that in $\mathrm{Mod}(K)$. □

Lemma 11 allows the proper definition of conceptual knowledge for all stable cognitive states.

Definition 23. (Conceptual Knowledge) The *conceptual knowledge* in a stable cognitive state K is denoted $\mathrm{Mod}(K)$ and is the inverse limit of the inverse system of models:

$$(\mathrm{Mod}(K'))_{K'\in K\downarrow}, (h_{K'',K'})_{K''\sqsubseteq K'\in K\downarrow}, (\psi_{K'',K'})_{K''\sqsubseteq K'\in K\downarrow})$$

6.3 Transcendental model

Remembering that $u_{\lim} \in D_0$ contains all the possible observations of a system, it is possible to define the model for transcendental logic given the possible observations and cognitions.

Definition 24. (Transcendental Model) Let D_0 be an object domain, $u_{\lim} \in D_0$ be the ideal containing on all possible observations, Σ be a signature of morphisms available for reasoning, and $\mathcal{X}$ be a denumerable set of variables. Then the *model for transcendental logic on* $(u_{\lim}, \Sigma)$ is the inverse limit of the K-inverse system of models with $K = (u_{\lim}, \mathrm{T}(\Sigma), \mathrm{T}(\Sigma) \setminus \mathrm{Repr}(\mathrm{T}(\Sigma)), R)$, with R the set pure representation terms in $\mathrm{T}(\Sigma \cup \mathcal{X})$.

Building on [30] this model is of practical interest, as the observed system can be partially specified with universally quantified implications between positive formulas, called geometric rules. In particular it was observed in [16] that deduction systems commonly employed for the symbolic analysis of cryptographic protocols, and in particular in static equivalence analysis, are examples of such theories.

7 Conclusion

The aim of this paper was to solve two challenges. A first one is to fill the gap in the formalisation of transcendental logic in [30] to precise how cognitions are achieved from experiences and cognitions, which is achieved in Definition 24. Having a first complete formalisation at hand to understand Kant's work opens the possibilities *(i)* of more concrete discussions on how to alter (or reject) that model to describe more precisely Kant's analysis of humans' thought process; *(ii)* of given a proper semantics to a natural language with names presented through examples; and *(iii)* interpreting the table of judgements of [14] as morphisms operating on different aspects a cognitive state. In particular the analysis of the relations between observations is not explored in this paper.

A second one was to provide a formal framework for anomaly detection in the context of intrusion detection systems. Until further work proves otherwise, modeling learning with morphisms on domains prevents the usage of most common Machine Learning algorithm. Despite this limitation, this setting is still applicable to at least some anomaly detection problems as it formalises [34]. In these cases we believe that it fills partially the gap described in [25] between what ML algorithms provide and what is actually needed for anomaly detection, and in particular the main recommendation of *understanding what the system does*.

We provide more details with references to the corresponding section of [25].

Based on the practical knowledge synthesis, *(III-A)* outlier detection can be achieved at least in some cases [34], and the *(B)* high cost of error can be addressed over time by Theorem 2. Conceptual knowledge is only the first step towards reducing the *(III-C)* semantic gap. Further work is needed to align the profusely created predicates of experiences with human-understandable concepts. In particular we believe that addition of object and representation terms in the cognitive state is driven by this process.

As it is a first-order model it is trivial to add rules (implications between geometric given the results of [30]) describing either normal behaviours or possible

attack scenarios. A missing part is the relation between these rules and the application of morphisms to check their validity. It is notable that dynamic proofs in [31] add facts to construct a model. When building a theory of a system, these facts have to be checked, *i.e.*, the terms and predicates of experience have to be constructed.

It fits the demands in *IV-D-2)* on result understanding as follows: Being complete lattice morphisms our learning algorithms all have an upper adjoint that provides a minimal explanation (as the meet of all viewed observations that lead to the current state) for each computed value. Ultimately the composition of these explanations provides an observation that explains the decision.

References

[1] Agrawal, R., Gunopulos, D., and Leymann, F. Mining process models from workflow logs. In Hans-Jörg Schek„ Gustavo Alonso„ Felix Saltor„ and Isidro Ramos„ editors, *Advances in Database Technology — EDBT'98*, pages 467–483, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[2] Andrej Bauer„ Lars Birkedal„ and Dana S. Scott,. Equilogical spaces. *Theor. Comput. Sci.*, 315(1):35–59, 2004.

[3] Andrew Chi-Chih Yao,. The Complexity of Pattern Matching for a Random String. *SIAM J. Comput.*, 8(3):368–387, 1979.

[4] AUTOSAR,. Specification of CAN Transport Layer. Technical Report 014, AUTOSAR, December 2017.

[5] Béatrice Longuenesse,. *Kant and the Capacity to Judge: Sensibility and Discursivity in the Transcendental Analytic of the Critique of Pure Reason*. Princeton University Press, Princeton, NJ, USA, 2001.

[6] Brian A. Davey, and Hilary A. Priestley,. *Introduction to Lattices and Order, Second Edition*. Cambridge University Press, 2002.

[7] Community Effort,. OBD-II Semantics of CAN Frames, 2019. Owners' assessment of the frame ID semantics in a Tesla Model 3 CAN Bus, available at https://docs.google.com/spreadsheets/d/1ijvNE4lU9Xoruvcg5AhUNLKr7xYyHcxa8YSkTxAERUw/edit#gid=0.

[8] Dana Angluin,. Learning Regular Sets from Queries and Counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.

[9] David I. Spivak, and Ryan Wisnesky,. On The Relational Foundations Of Functorial Data Migration. *CoRR*, abs/1212.5303, 2012.

[10] Deepak Kapur, and Paliath Narendran,. The Knuth-Bendix Completion Procedure and Thue Systems. *SIAM J. Comput.*, 14(4):1052–1072, 1985.

[11] Gergei Bana, and Hubert Comon-Lundh,. A Computationally Complete Symbolic Attacker for Equivalence Properties. In Gail-Joon Ahn„ Moti Yung„ and Ninghui Li„ editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 609–620. ACM, 2014.

[12] Gottlob Frege,. *Begriffsschrift: Eine der Arithmetischen Nachgebildete Formelsprache des Reinen Denkens*. Halle a.d.S.: Louis Nebert, 1879.

[13] Gunawardena, J. Deducing causal relationships in CCS. In E.Veni Madhavan, C., editor, *Foundations of Software Technology and Theoretical Computer Science*, pages 161–170, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[14] Immanuel Kant,. *Critique of the Pure Reason ; Translated from the German by Paul Guyer and Allen W. Wood*. Cambridge University Press, 1998.

[15] ISO,. Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2: Transport protocol and network layer services. Technical Report 15765-2, ISO, 2016.

[16] Joshua D. Guttman,. Establishing and preserving protocol security goals. *J. Comput. Secur.*, 22(2):203–267, 2014.

[17] Laraba, A., François, J., Chowdhury, S. R., Chrisment, I., and Raouf Boutaba,. Mitigating TCP Protocol Misuse With Programmable Data Planes. *IEEE Trans. Netw. Serv. Manag.*, 18(1):760–774, 2021.

[18] Leslie G. Valiant,. A Theory of the Learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[19] Mark Gold, E. Complexity of automaton identification from given data. *Information and Control*, 31:302–320, 1978.

[20] Mark V. Lawson,. 1. Classical Stone Duality. available at `http://www.macs.hw.ac.uk/~markl/1-stone-duality.pdf`, 2018.

[21] Martín Abadi, and Véronique Cortier,. Deciding Knowledge in Security Protocols Under Equational Theories. In Josep Díaz„ Juhani Karhumäki„ Arto Lepistö„ and Donald Sannella„ editors, *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, volume 3142 of *Lecture Notes in Computer Science*, pages 46–58. Springer, 2004.

[22] Nachum Dershowitz, and Jean-Pierre Jouannaud,. Rewrite Systems. In Jan van Leeuwen„ editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 243–320. Elsevier and MIT Press, 1990.

[23] Nuel D. Belnap Jr.,. A Useful Four-Valued Logic. In *Modern Uses of Multiple-Valued Logic*, pages 5–37. D. Reidel Publishing Company, 1977.

[24] Phokion G. Kolaitis„ Lucian Popa„ and Kun Qian,. Knowledge Refinement via Rule Selection. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 2886–2894. AAAI Press, 2019.

[25] Robin Sommer, and Vern Paxson,. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316, 2010.

[26] Rudolf Wille,. Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In Rival, I., editor, *Ordered Sets*, volume 83 of *NATO Advanced Study Institutes Series*, pages 445–470. Springer Netherlands, 1982.

[27] Samson Abramsky,. Domain Theory in Logical Form. *Ann. Pure Appl. Log.*, 51(1-2):1–77, 1991.

[28] Samson Abramsky,. Domain Theory. In Samson Abramsky„ Dov M. Gabbay„ and Maibaum, T. S. E., editors, *Handbook of logic in computer science. Volume 3. Semantic Structures*. Clarendon Press, 1994.

[29] Steven Vickers,. Continuity and Geometric Logic. *J. Appl. Log.*, 12(1):14–27, 2014.

[30] Theodora Achourioti, and Michiel van Lambalgen,. A Formalization of Kant's Transcendental Logic. *Rev. Symb. Log.*, 4(2):254–289, 2011.

[31] Thierry Coquand,. A Completeness Proof for Geometric Logic. available at https://www.cse.chalmers.se/~coquand/site.pdf.

[32] Uriel Feige„ Amos Fiat„ and Adi Shamir,. Zero Knowledge Proofs of Identity. In Alfred V. Aho„ editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 210–217. ACM, 1987.

[33] Wardell, J. Log of a Tesla Model 3 Car, 2019. available at https://www.dropbox.com/s/vpz5b0c78qmqlt8/Model3Log2019-10-02v10.asc.zip?dl=0&file_subpath=%2FModel3Log2019-10-02v10.asc.

[34] Yannick Chevalier,. Data Exchange for Anomaly Detection: The Case of the CAN Bus. In *Conference on Artificial Intelligence for Defense (CAID 21)*, pages 25–32, Rennes, France, November 2021. DGA : Direction Générale de l'Armement - Ministère français des Armées.

[35] Yannick Chevalier, and Michaël Rusinowitch,. Compiling and Securing Cryptographic Protocols. *Inf. Process. Lett.*, 110(3):116–122, 2010.

[36] Yannick Chevalier, and Michaël Rusinowitch,. Implementing Security Protocol Monitors. In Temur Kutsia„ editor, *Proceedings of the 9th International Symposium on Symbolic Computation in Software Science, SCSS 2021, Hagenberg, Austria, September 8-10, 2021*, volume 342 of *EPTCS*, pages 22–34, 2021.

A Example application: Detection of Network Protocols

A.1 Detection of purely parallel processes

While process mining [1] usually focuses on the discovery of workflow graphs encoding conditionals and loops, the detection of synchronisation between messages is more often, as *e.g.*, in the case of a protocol, a sequential activity without holes: Messages a and b in an order are always followed by messages c and d, etc. In the case of the CAN Bus these sequences occur in two contexts. In a MFM protocol execution that starts with an initial message, replied to with an ack, and followed by a sequence of messages with a counter. Or in the case of a proper design of the CAN Bus in which such synchronisation is introduced to avoid collisions between messages. These are examples of *purely parallel* processes [13] that are built with sequential and parallel composition only, *i.e.*, with no non-deterministic choice.

Let S be a DXS. A subset $S_c \subseteq S$ of terms which are applications of split functions on "X" is called a *pre-classification*. If it is maximal then it is the classification of the DXS. If the codomain of "X" is the set of possible CAN frames, then the codomain of each t in the classification S_c is also the set of CAN frames. We assume that all the events in the log are distinct *e.g.*, by adding a time of capture. To simplify notations we denote $e \in u$ if the event e occurs at some position in the word u, and $|u|$ the length of the word u.

Algorithm. Let S be a DXS with a classification set S_c. A representation function nb_occ is applied on each $t \in S_c$ to count the number of occurrences of messages in the interpretation of the term t. The state of the network after seeing a trace u is represented by a boolean square matrix$P_{S_c}(u)$ indexed by terms in S_c and defined inductively on traces with:

$$P^u_{S_c}(t,t') = \begin{cases} 1 & \text{If } u = \varepsilon \\ P^v_{S_c}(t,t') \wedge (\text{nb\_occ}(t') \neq \text{nb\_occ}(t') & \\ P^u_{S_c}(t,t') & \text{Otherwise} \end{cases}$$

When analyzing a log $\Sigma_{u\in U} u$ we let

$$P^{\Sigma_{u\in U} u}_{S_c} = \Sigma_{u\in U} P^u_{S_c}$$

be the coordinate-wise conjunctions of the matrice $P^u_{S_c}$ for $u \in U$. Ordering the matrice with the extension on all coordinates of the ordering on booleans, one easily checks that the conjunction being the sup of boolean values, this function is monotonic and algebraic and thus can be extended by continuity to infinite words.

In the following definition, the semi-colon denotes sequential composition, a set of term denotes a parallel composition of these terms, and the while *true* do denotes an unbounded iteration of the process.

Definition 25. (Purely Parallel Process—PPP Let S be a DXS with a classification set S_c. An *iterated purely parallel process* is a process of the while *true* do $(A_1; \ldots; A_n)$ where $A_1, \ldots, A_n$ is a partition of a subset of S_c.

A PPP $P =$ while *true* do $(A_1; \ldots; A_n)$ is *unambiguous* on trace $u = (e_i)_{1\leq i\leq N}$ if furthermore for all $1 \leq i \leq N$ if e_i occurs in $[t]_u$ and $intput'$ for $t, t' \in \cup_{i=1}^n A_i$ then $t = t'$.

Lemma 12. *Assume the purely parallel process* $P =$ while *true* do $(A_1; \ldots; A_n)$ *is executed on the network, that the trace* u *is observed, and that* P *is unambiguous for* u*. Then for all* $1 \leq i < j \leq n$ *and for all* $t' \in A_i, t \in A_j$ *we have:*

$$|[t']_u| = |[t]_u| \text{ or } |[t']_u| = |[t]_u| + 1$$

Proof. We first prove the following statement that is satisfied by the less constrained process:

$$P = \text{while } \textit{true} \text{ do } (\cup_{i=1}^{n} A_i)$$

Claim. For every trace u there exists L_u, H_u, and E_u such that:

- L_u, H_u is a partition of $\cup_{i=1}^{n} A_i$ and $L_u \neq$;
- $E_u \subseteq \cup_{i=1}^{n} A_i$ is the subset of the terms of the process that have not been observed in the current iteration of the process;
- The following equalities are satisfied:

$$\begin{cases} \forall t, t' \in L_u, & |[t]_u| = |[t']_u| & (A) \\ \forall t, t' \in H_u, & |[t]_u| = |[t']_u| & (B) \\ \forall t \in L_u, \forall t' \in H_u, & |[t]_u| + 1 = |[t']_u| & (C) \\ E_u = L_u & & (D) \end{cases}$$

Proof of the claim. By induction on the length of the trace u.

- If $u = \varepsilon$ let $H_\varepsilon = \emptyset$ and $L_\varepsilon = E_\varepsilon = \cup_{i=1}^{n} A_i$. Since $H_u = \emptyset$ the equations (C) and (D) are trivially satisfied. The interpretation is strict so the equation (A) is satisfied (and the length is equal to 0). The equation (D) is satisfied by definition.
- Assume the claim stands for a trace u and consider the trace $u \cdot e$.
  - If $e \notin \cup_{i=1}^{n} \cup_{t \in A_i} [t]_{u \cdot e}$, *i.e.*, the observed event e is not associated with a term in the process, then setting $L_{u \cdot e} = L_u$, $H_{u \cdot e} = H_u$, and $E_{u \cdot e} = E_u$ satifies the constraints of the claim;
  - Otherwise there exists $t \in \cup_{i=1}^{n} A_i$ such that $e \in [t]_{u \cdot e}$. Since the process is unambiguous the term t has not been observed in the current iteration of the loop, and thus $t \in E_u$. By induction this implies $t \in L_u$. One then easily checks that the equalities are all satisfied by setting:
    - If $L_u = \{t\}$: $L_{u \cdot e} = E_{u \cdot e} = \cup_{i=1}^{n} A_i$ and $H_{u \cdot e} = \emptyset$;
    - Otherwise $L_{u \cdot e} = L_u \setminus \{t\}$, $E_{u \cdot e} = L_{u \cdot e}$, and $H_{u \cdot e} = H_u \cup \{t\}$;

□

Claim. For any trace u, there exists $1 \leq i_t \leq n$ such that $H_u \subseteq \cup_{i=1}^{i_t} A_i$, $L_u \subseteq \cup_{i=i_t}^{n} A_i$, and if $E_u \neq E_{u \cdot e}$ there exists $t \in L_u \cap A_{i_t}$ such that $e \in [t]_u$.

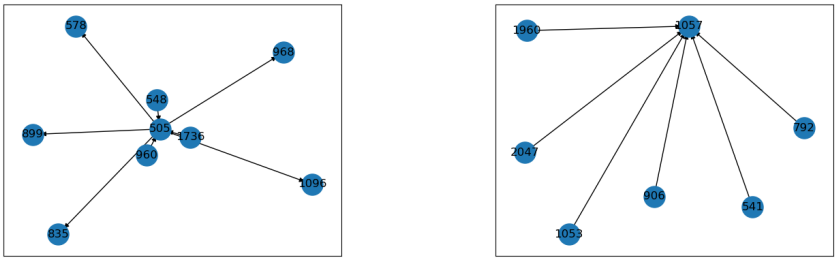
The proof is a copy of the one of the first claim, but it has been separated for clarity. One has to observe that to respect the sequentiality of the process, when $t \in E_u \cap A_j$ is chosen we must have $E_u \cap \cup_{i=1}^{j-1} A_i = \emptyset$.

The Lemma follows from the two claims by case analysis on the repartition of t, t' in H_u and L_u using the second claim to transform it into a case analysis on the indice. □

Proposition 10. *Let S be a DXS with a classification set S_c, and u be a trace. Assume the purely parallel process $P =$* while *true* do $(A_1; \ldots; A_n)$ *is executed on the network, and that P is unambiguous for u. Then for every $t \in A_i, t' \in A_j$ with $i > j$ we have $P^u_{S_c}(t, t') = 1$.*

The other direction cannot be proven as a purely parallel process can be found by coincidence when there are few iterations of the loop.

We present in Fig. 1 examples of the analysis result on a Model 3 CAN Bus log [33] representing 418s of execution and 879682 frames.



(a) The star PPP is the most common one on the Tesla Model 3: Two sets of *id*s have one of the set of nodes which is separated by a single node. This PPP is iterated 4181 times in the log.

(b) A degeneracy of the star PPP is to have one of the set of nodes which is empty. This PPP is also iterated 4181 times in the log.

Figure 1: Examples of purely parallel processes found on a Tesla Model 3 log. In total 29 CAN frames *id* were isolated, 32 were discovered in a request-response process, and 126 involved were in a star process either proper as in Fig. 1a or degenerate as in Fig. 1b.

A.2 Discovering Protocol Instances

Setting.. A *protocol* is a program running among multiple participants in a network and observed through the messages exchanged by these participants. It is specified with a set of *roles* that are programs and the execution of a role is a process called an *actor*. In addition to the control rules defined in the role an actor is playing, the actor has a local memory that is updated while running the protocol. We note that as in [36] under this definition a multi-step attack in which an adversary communicates with different actors on the network is also a protocol.

As in [17] each role is modeled with an *Extended Finite State Machine* (EFSM), a finite state automaton enriched with rules updating the state of an actor playing that role. This automaton is *deterministic* in the sense that an actor playing a role and receiving a message can either reject that message as non-conforming with the protocol specification and leaves its state unchanged, or change its state according to the rules set in the rule in exactly one possible way. Thus to each state of the automaton we associate a couple of functions (g, p) where g returns whether a message received in the current participant's state is acceptable, and p that performs the update of the memory and of the automaton state. A group of actors playing the roles defined in the protocol and communicating one with another is called a *session*.

More formally let X be a finite set of sorted variables, $\mathcal{R}$ be a finite set of

constants (the roles), $\mathcal{Q}$ be a finite state of automaton states, and $\mathcal{E}$ be the set of the possibly observed events (the messages). A *memory state* is a ground substitution of domain X. We denote $\mathcal{M}$ the set of possible memory states. A *state* is a couple $(q, \sigma) \in \mathcal{Q} \times \mathcal{M}$. A *guard* is a function $\mathcal{Q} \times \mathcal{M} \times \mathcal{E} \to \mathcal{B}$. A *post-transition* is a function $\mathcal{Q} \times \mathcal{M} \times \mathcal{E} \to \mathcal{Q} \times \mathcal{M} \times \mathcal{E}$. If p is a transition function and $f(q, m, e)) = (q', m', e')$ then upon receiving the message e, the actor with a state (q, m) updates it to (q', m') and sends the message e'.

An *EFSM* E is a set of tuples $\{(q, g_q, f_q)\}_{q \in \mathcal{Q}_r \subseteq \mathcal{Q}}$ where g_q is a guard and f_q is a post-transition function. A *protocol* P is a finite set of tuples $\{(r, \iota_r, E_r)\}_{r \in \mathcal{R}}$ where each E_r is an EFSM, $\iota_r \in \mathcal{Q}$ is the *initial state* of the role r, and if $r \neq r'$ then $\mathcal{Q}_r \cap \mathcal{Q}_{r'} = \emptyset$. The bin role contains one state $\mathcal{Q}_b = \{\iota_b\}$, and one rule (ι_b, g_b, f_b) where the guard g_b is always true and f_b does nothing.

Algorithm.. Again we build a representation of the system inductively on a trace u. Let $P = \{(r, \iota_r, E_r)\}_{r \in \mathcal{R}}$ be a protocol. Participants are identified uniquely and are assumed to always or never play a role of the protocol. Depending on the network type they can be identified by a socket address or in the case of the CAN bus by the *id* of the frame. Actors are participants playing a role in the protocol. We first build a derived object simulating the network after observing a trace u Our representation of the network is based on an internal state $\mathcal{A}_u$ of potential actors with their state. Accordingly we let:

$$\mathcal{A}(\varepsilon) = \{(\iota_{p,r}, m_{p,r}, E_{p,r})\}_{(p,r) \in \mathcal{P} \times \mathcal{R}}$$

assuming initially every participant may play a role in the protocol.

Inductively the internal state of the function is defined as follows:

- If there exists $(q_{p,r}, m_{p,r}, E_{p,r}) \in \mathcal{A}(u)$, a transition $(q_{p,r}, g_{q_{p,r}}, f_{q_{p,r}}) \in E_{p,r}$, and an event $e' \in u$ such that:

$$\begin{cases} g_{q_{p,r}}(e', m_{p,r}) & = & 1 \\ f_{q_{p,r}}(e', q_{p,r}, m_{p,r}) & = & (e, q'_{p,r}, m'_{p,r}) \end{cases}$$

  Then:

$$\begin{aligned} \mathcal{A}(u \cdot e) = (\mathcal{A}(u) & \setminus \{(q_{p,r}, m_{p,r}, E_{p,r})\}) \\ & \cup \left\{(q'_{p,r}, m'_{p,r}, E_{p,r})\right\} \end{aligned}$$

- Otherwise the participant has sent a message that is not conformant with the protocol specification, and thus per our assumption never is an actor, in any role, of that protocol:

$$\mathcal{A}(u \cdot e) = \mathcal{A}(u) \setminus \{(q_{p,r}, m_{p,r}, E_{p,r}) \in \mathcal{A}(u) \,|\, r \in \mathcal{R}\}$$

The representation of the network after observing the trace u is the subset of participants p such that there exists an EFSM indexed by p in $\mathcal{A}(u)$. We order these sets with $A \sqsubseteq B$ if and only if $B \subseteq A$. The top element is the emptyset and the bottom element is the set of all participants, and the join $A \sqcup B$ of two sets is their intersection. This domain clearly is a representation domain. The computation of $\mathcal{A}(u)$ is extended to worlds with $\mathcal{A}(\Sigma_{u \in U} u) = \bigsqcup_{u \in U} \mathcal{A}(u)$, and is clearly continuous. Though it possible for a participant to play different roles concurrently and be correctly detected, playing concurrently the same role may (as in the case of a server in [17]) make the participant appear as not playing along the rules of the protocols. The benefit of this simplification is the low computational and memory cost in practice.

Our implementation of the Multi-Frame Message protocol [15] has correctly detected on the Tesla Model 3 log the similar CAN-TP [4] protocol according to owners' common findings [7].

Institut de Recherche en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

ASR – Architecture, Systems and Networks

RMESS – Networks, Mobile, Embedded, Wireless, Sattelites

SEPIA – Operating systems, distributed systems, from Middleware to Architecture

SIERA – Service IntEgration and netwoRk Administration

T2RS – Real-Time in networks and systems

TRACES – Trace stands for research groups in architecture and compilation for embedded systems

CISO – HPC, Simulation, Optimization

APO – Parallel Algorithms and Optimisation

REVA – Real Expression Artificial Life

FSL – Reliability Systems and Software

ACADIE – Assistance for certification of distributed and embedded applications

ARGOS – Advancing Rigorous Software and System Engineering

ICS – Interactive Critical Systems

SM@RT – Smart Modeling for softw@re Research and Technology

GD – Data Management

IRIS – Information Retrieval and Information Synthesis

PYRAMIDE – Dynamic Query Optimization in large-scale distributed environments

SIG – Generalized information systems

IA – Artificial Intelligence

ADRIA – Argumentation, Decision, Reasoning, Uncertainty and Learning methods

LILaC – Logic, Interaction, Language and Computation

MELODI – Methods and Engineering of Language, Ontology and Discourse

ICI – Interaction, Collective Intelligence

ELIPSE – Human computer interaction

SMAC – Cooperative multi-agents systems

TALENT – Teaching And Learning Enhanced by Technologies

SI – Signals and Images

MINDS – coMputational Imaging anD viSion

SAMoVA – Structuration, Analysis, Modeling of Video and Audio documents

SC – Signal and Communications

STORM – Structural Models and Tools in Computer Graphics

TCI – Images processing and understanding