Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems Submitted 9/2021; published 11/2021

Signature Entrenchment and Conceptual Changes in Automated Theory Repair

Xue Li[1] XUE.SHIRLEY.LI@ED.AC.UK
Alan Bundy A.BUNDY@ED.AC.UK
Eugene Philalithis E.PHILALITHIS@ED.AC.UK
School of Informatics, University of Edinburgh, UK

Abstract

Human beliefs change, but so do the concepts that underpin them. The recent Abduction, Belief Revision and Conceptual Change (ABC) repair system combines several methods from automated theory repair to expand, contract, or reform logical structures representing conceptual knowledge in artificial agents. In this paper we focus on conceptual change: repair not only of the membership of logical concepts, such as what animals can fly, but also concepts themselves, such that birds may be divided into flightless and flying birds, by changing the *signature* of the logical theory used to represent them. We offer a method for automatically evaluating entrenchment in the signature of a Datalog theory, in order to constrain automated theory repair to succinct and intuitive outcomes. Formally, *signature entrenchment* measures the inferential contributions of every logical language element used to express conceptual knowledge, i.e., predicates and the arguments, ranking possible repairs to retain valuable logical concepts and reject redundant or implausible alternatives. This quantitative measurement of signature entrenchment offers a guide to the plausibility of conceptual changes, which we aim to contrast with human judgements of concept entrenchment in future work.

1. Introduction

Knowledge in artificial agents is classically modelled via the medium of axioms. Stores of assertions and relations between them, expressed in a logical theory using a predefined language of *predicates* and *constants*, represent agents' grasp of their world. Together, the definitions of these predicates and constants comprise the *signature* of a logical theory. Yet neither axioms nor signatures are static: when the world changes and causes faults in the theory, they must change too. Automated theory repair focuses on faults - errors of formal reasoning - as drivers for theory change (Bundy, 1983).

In human agents, beliefs also change, and so do the concepts that underpin them. Whether seen as prototypes (Posner & Keele, 1968), or exemplars (Medin & Schaffer, 1978), or causal theories (Gopnik & Wellman, 1992), the boundaries of concepts are continually reassessed based on novel input. But not all concepts are equal: e.g. basic-level categories that more sharply distinguish their members from other objects, such as 'car', are more salient than vaguer categories such as 'vehicle'

1. The first and second authors are supported by Huawei grant CIENG4721/LSC and the second author by UKRI grant EP/V026607/1.

(Murphy & Brownell, 1985; Rosch et al., 1976). Alternative concepts for the same individual object can have different worth, based on the quality of inferences they permit or deny.

In this paper we discuss an initial attempt at reconciling these two types of changes, artificial and natural, in the context of conceptual knowledge representation. We apply an automated theory repair algorithm, the Abduction, Belief Revision and Conceptual Change [ABC] System (Li, 2021; Li et al., 2018) to flexibly revise an agent's prior knowledge of simple concepts in response to novel input. ABC combines operations based on belief revision (Alchourrón et al., 1985) and abduction (Frankfurt, 1958) with a newer algorithm for logical signature repair (Bundy & Mitrovic, 2016). The resulting system automatically modifies conceptual knowledge in two ways: (a) it automatically adds or deletes axioms - such as assertions about concept membership, e.g. *Tweety is a penguin*, or relations between concepts, e.g. *all penguins are birds* - and (b) it automatically changes *concepts themselves*, by editing the logical language of predicates and constants, known as the signature, used to express them.

This flexibility comes at a cost. The number of all possible repairs creates a problem of scale: both because computing them all can become computationally infeasible, and because many of these repairs can distort logical concepts in unintuitive ways, compared to how human concepts (vs. purely logical structures) are adapted. To control and direct changes to conceptual knowledge, including concepts themselves, we utilise the notion of *entrenchment*. Entrenchment is a popular cognitive notion, most often used in connection with the role of memory (Schmid, 2016), or of predictive bias in language use (Pickering & Garrod, 2013); but also in the broader sense we favour here, as a putative measure of conceptual 'staying power', e.g. evidenced in basic-level categories (Murphy & Brownell, 1985). The notion of entrenchment is also popular in belief revision, standing in for information value: the more entrenched, the more valuable a belief, and the less inclined a system should be to change it. Some desired properties of entrenchment for beliefs have been proposed by Gärdenfors (1988), but these do not assume one can quantitatively measure entrenchment, because the factors that affect entrenchment are diverse, and their interactions can be complicated. The work we present in this paper thus covers new ground for automated reasoning about concepts.

Presently, we consider the specific case of *signature entrenchment*: the value, or staying power, of a concept in itself. We discuss requirements for adding signature entrenchment to ABC, then demonstrate and evaluate its implementation in the form of a graphical meta-theory. In line with the cognitive notion of entrenchment as salience, our meta-theory captures the individual contributions of concepts to inferences, while retaining the formal notion of entrenchment as information value.

The paper is structured as follows: first off, we briefly introduce the ABC repair system and our hypothesis in §2. Then we define signature repair, under the logical term 'conceptual change', in §3. Then we present our main achievement, the measurement of entrenchment for signature elements, i.e. predicates and their arguments[2] in §4-§4.2. Finally, we evaluate the performance of ABC combined with our signature entrenchment measure in §5, before concluding remarks in §6.

2. Code for all the work in this paper is available on GitHub: https://github.com/XuerLi/Publications/tree/main/ACS2021.

2. ABC Repair System and Hypothesis

ABC's representations of conceptual knowledge takes the form of Datalog theories consisting of sets of axioms. Axioms in Datalog theories are Horn clauses. We use the Prolog convention that variables start with uppercase and constants and predicates start with lowercase. We will represent these clauses using *Kowalski Form*.

$$Q_1 \wedge \ldots \wedge Q_m \implies P_1 \vee \ldots \vee P_n$$

where the Q_j and P_i are propositions. In Horn clauses, n is either 0 or 1. In Datalog, the arguments of the propositions are either constants or variables, i.e., there are no non-nullary functions. This makes SL resolution with a fair search strategy a decision procedure for Datalog theories. This is important because, when diagnosing faults, ABC needs to be sure whether conjectures are theorems.

- If $m = 0$ and $n = 1$ the clause is an *assertion*.
- If $m > 0$ and $n = 1$ the clause is a *rule*.
- If $m > 0$ and $n = 0$ the clause is a *goal*.
- If $m = 0$ and $n = 0$ the clause is *empty*.

Example 2.1 shows a Datalog theory. Note the $\implies$ arrow is retained even when $m = 0$ or $n = 0$.

Example 2.1. *Motherhood Theory* $\mathbb{T}_m$.

$\implies mum(lily, victor)$	$\implies mum(lucy, tom)$
$\implies mum(anna, david)$	$mum(X,Y) \wedge mum(Z,Y) \implies X = Z$
$\implies mum(anna, victor)$	$mum(X,Y) \implies families(X,Y)$

As we further discuss below in §3, this is a theory of motherhood relations. In a logical theory, conceptual knowledge is stored in a language of predicates and constants specified in its signature, as defined below in 2.1. $\mathbb{T}_m$'s signature includes the predicates $mum/2$, $=/2$, and $families/2$ as well as the constants $lily$, $victor$, $anna$, $david$, $lucy$, and tom. Here p/n means that predicate p has n arguments, such that $mum/2$ has two arguments, a mother and a child.

Definition 2.1 (Signature)**.** *A signature is the grammar and the corpus of the language in which a logical theory is written. For a Datalog theory, the signature contains the following elements:*

Predicates: *a predicate string maps individuals to truth values, true or false, according to whether they satisfy that predicate.*

Predicate Arity: *a predicate's arity is the number of its arguments, which is a non-negative integer.*

Constants: *a constant string stands for an individual.*

In ABC, a store of environmental observations are the benchmark of the input Datalog theory's correctness. These observations are represented by a *preferred structure* $\mathbb{PS}$, consisting of a pair of sets of ground propositions: those propositions explicitly observed to be true $\mathcal{T}(\mathbb{PS})$ and those explicitly observed to be false $\mathcal{F}(\mathbb{PS})$. Selected Literal Resolution (SL) (Kowalski & Kuehner, 1971) is applied to the input DataLog theory, to prove theorems. When these theorems conflict with $\mathbb{PS}$, incompatibility and insufficiency faults, defined below, are detected. Based on the proofs, or failed proofs, of these faults, ABC automatically repairs the input Datalog theory.

Definition 2.2 (Types of Fault)**.** *Let $\mathbb{T}$ be a Datalog theory.*

Incompatibility: $\exists\phi.\ \mathbb{T} \vdash \phi \wedge \phi \in \mathcal{F}(\mathbb{PS})$*;*

Insufficiency: $\exists\phi.\ \mathbb{T} \nvdash \phi \wedge \phi \in \mathcal{T}(\mathbb{PS})$

$\mathbb{PS}$ is not required to contain the whole set of potentially relevant true or false statements about an observed world. An incomplete $\mathbb{PS}$ can provide a partial picture, which is often sufficient and practicable. ABC repairs faulty theories using ten *repair operations*, five of them for repairing incompatibilities, and five for repairing insufficiencies. They are formally given in Definitions 2.3 and 2.4, respectively.

Definition 2.3 (Repair Operations for Incompatibility)**.** *In the case of incompatibility, the unwanted proof can be blocked by causing any of the resolution steps to fail. Suppose the targeted resolution step is between a goal, $P(s_1, \ldots, s_n) \implies$, and an axiom, $Body \implies P(t_1, \ldots, t_n)$, where each s_i and t_i pair can be unified. Possible repair operations are as follows:*

Belief Revision 1: *Delete the targeted axiom: $Body \implies P(t_1, \ldots, t_n)$.*

Belief Revision 2: *Add an additional precondition to the body of an earlier rule axiom which will become an unprovable subgoal in the unwanted proof. To be effective, this precondition must share at least one variable with another precondition.*

Reformation 1: *Rename P in the targeted axiom to either a new predicate or a different existing predicate P'.*

Reformation 2: *Increase the arity of all occurrences of P in the axioms by adding a new argument. Wlog, assume it is the last. Ensure that the new arguments, s_{n+1} and t_{n+1}, in the targeted occurrence of P, are not unifiable. In Datalog, this can only be ensured if they are unequal constants at the point of unification.*

Reformation 3: *For some i, suppose s_i is C. Since s_i and t_i unify, t_i is either C or a variable. Change t_i to either a new constant or a different existing constant C'.*

Definition 2.4 (Repair Operations for Insufficiency)**.** *In the case of insufficiency, the wanted but failed proof can be unblocked by causing a currently failing resolution step to succeed. Suppose the chosen resolution step is between a goal $P(s_1, \ldots, s_m) \implies$ and an axiom $Body \implies P'(t_1, \ldots, t_n)$, where either $P \neq P'$ or $P = P'$, so $m = n$, but for some i, s_i and t_i cannot be unified. Possible repair operations are:*

Abduction 1: *Add a new axiom whose head unifies with the goal $P(s_1, \ldots, s_m)$ by analogising an existing rule or formalising a precondition based on a theorem whose arguments overlap with the ones of that goal.*

Abduction 2: *Locate the rule axiom whose body proposition created this goal and delete this proposition from the axiom.*

Reformation 4: *Replace $P'(t_1, \ldots, t_n)$ in the axiom with $P(s_1, \ldots, s_m)$.*

Reformation 5: *Suppose s_i and t_i are not unifiable. Decrease the arity of all occurrences P' by one by deleting its i^{th} argument.*

Reformation 6: *If s_i and t_i are not unifiable, then they are unequal constants, say, C and C'. Either (a) rename all occurrences of C' in the axioms to C or (b) replace the offending occurrence of C' in the targeted axiom by a new variable.*

Reformation thus changes the signature of a theory in three general ways: (a) renaming predicates or arguments, (b) adding or deleting arguments, (c) switching between variables and constants.

With overproduction of repairs being a common issue for this family of algorithms (Gärdenfors, 2003; Urbonas et al., 2020; Li, 2021). ABC aims to rank all possible repairs from most to least preferred based on the properties in Definition 2.5.

Definition 2.5 (Preferred Repairs)**.** *Preferred repairs are those repaired Datalog theories with the following properties:*

1. *their repairs satisfy the preferred structure;*
2. *the repair operations applied to generate the theories are all necessary to repair faults;*
3. *their repairs are intuitive.*

The first two properties for preferred repairs can be evaluated formally. The final property reflects a heuristic measure of human judgement on which alternative concepts are more intuitive than others. As ABC cannot make this judgement automatically, this aspect of its performance is evaluated by human users, to select the most intuitive among all produced repairs. Though we are not measuring this empirically here, the contrast with human judgements in an experimental setting provides an obvious avenue for future work. We revisit this point below in §6.

In this paper, we further define and measure signature entrenchment within the framework of the ABC system (Li, 2021). Signature entrenchment is used to rank the repairs that involve signature changes, shown in Figure 1. Our claim, and formal hypothesis about the ABC system's performance when incorporating signature entrenchment, is given below, and then revisited and evaluated in §5.

Definition 2.6 (**Hypothesis**)**.** *Automatically ranking signature repair outcomes by the ABC system using signature entrenchment scores will* reliably prioritise preferred repairs *(as per Definition 2.5).*

In other words, we claim that signature repair can be constrained by an automatic measure of signature entrenchment, such that ABC prioritises repairs that are consistent with a provided store of observations, that make no unnecessary changes, and that would be judged as more intuitive by a human agent.

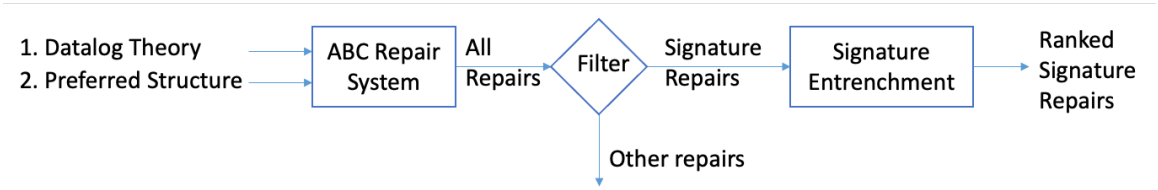

Figure 1. Signature entrenchment ranks repairs by the ABC system that involve signature changes. The ABC system takes two inputs, a Datalog theory and a preferred structure, and outputs repaired theories by ranking.

3. Conceptual change

'Conceptual change' is our term for signature repair: a change in the definition of logical concepts, rather than to their contents, equivalent to a concept such as 'cup' dividing into more concepts such as 'glass' and 'mug' - or a 'car' with siginficant storage space splitting off to become a 'van'. This section defines and discusses each kind of conceptual change that ABC can make, using the Datalog theory $\mathbb{T}_m$ previously introduced in Example 2.1, which we replicate again below for convenience.

Example 2.1 Motherhood Theory $\mathbb{T}_m$.

$\Longrightarrow mum(lily, victor)$	$\Longrightarrow mum(lucy, tom)$
$\Longrightarrow mum(anna, david)$	$mum(X,Y) \wedge mum(Z,Y) \Longrightarrow X = Z$
$\Longrightarrow mum(anna, victor)$	$mum(X,Y) \Longrightarrow families(X,Y)$

In $\mathbb{T}_m$, the concept of motherhood is given by the predicate $mum/2$, which indicates there are two components needed to define the logical concept of being a mother: the mother and her child. Argument constants are instances (or exemplars) of mothers, e.g., $anna$ is the mum of $david$. The first rule states that someone can only have one mum. The second rule states that each mum and her children must all belong to the same family.

A predicate and its arguments are crucial for formalising the corresponding concept: the former names a concept, and the latter gives the components (i.e. constants) that are related by that concept in a certain order. The set of potential constants for each argument is its *argument domain*.

Definition 3.1 (Argument Domain)**.** *In a logical theory $\mathbb{T}$, the domain of an argument w.r.t. a predicate is the set of all constants that can appear in place of that argument in the theorems of the theory. The function $\mathcal{D}(p,\ n, \mathbb{T})$ returns the argument domain of the n_{th} argument of predicate p in $\mathbb{T}$.*

$$\mathcal{D}(p, i, \mathbb{T}) = \begin{cases} \{c_i | \mathbb{T} \vdash p(\vec{c}),\ c_i = \gamma(p(\vec{c}), i)\} & 1 \leq i \leq arity(p) \\ \emptyset & otherwise \end{cases} \tag{1}$$

where $\gamma(p(\vec{c}), i) = \gamma(p(c_1, c_2, ...c_n), i) = c_i$, and $c_1, c_2..c_n$ are constants.

Accordingly, the domain of the first argument of predicate $mum/2$ in the above $\mathbb{T}_m$ is $\mathcal{D}(mum, 1, \mathbb{T}_m) = \{anna,\ lily,\ lucy\}$. According to $\mathbb{T}_m$ there are thus three mothers: Anna,

Lily and Lucy. When a constant is eliminated or a new constant discovered, the corresponding argument domain is retracted or expanded respectively.

Resembling operations from belief revision (Alchourrón et al., 1985), a logical concept in a Datalog theory can be changed in three main ways: expansion, contraction and revision.

Definition 3.2 (Logical Conceptual Changes)**.**

Conceptual Expansion: *A new predicate/constant is added to the signature.*

Conceptual Contraction: *An existing predicate/constant is removed from the signature.*

Conceptual Revision: *The arity of a predicate is changed; or constants are split or merged in the signature.*

ABC employs Reformation, Bundy & Mitrovic (2016), a domain-independent automatic repair algorithm, for the implementation of conceptual change. The change of arity in Conceptual Revision corresponds to the change of the numbers of the components that constitute the logical concept which the predicate represents. The argument domain can be changed when constants are split or merged. Different argument domains refer to different sets of instances (or exemplars) that belong under that concept, such as Lucy, Lily and Anna as mothers in $\mathbb{T}_m$.

Arity change is necessary when the number of the components of a logical conceptual changes. For example, when a new variant of a category is discovered, and the existing arity cannot describe it, we need to increase the arity of the corresponding predicate. In $\mathbb{T}_m$, Lily and Anna are both Victor's mother. If we take Lily to be Victor's birth mother, then Anna is the *stepmother* of Victor. For distinguishing these different types of mothers, a new argument could be added to assign the motherhood variant. The revised logical concept mum, and the corresponding theory $\mathbb{T}_{rm}$ that more correctly represents the mother category, is shown in Example 3.1, with changes highlighted.

Example 3.1. *Enriched Motherhood Theory $\mathbb{T}_{rm}$ with Conceptual Changes.*

$$\begin{array}{l|r} \implies mum(lily, victor, \mathit{birth}) & \implies mum(lucy, tom, \mathit{birth}) \\ \implies mum(anna, david, \mathit{birth}) & mum(X, Y, \mathit{birth}) \wedge mum(Z, Y, \mathit{birth}) \implies X = Z \\ \implies mum(anna, victor, \mathit{step}) & mum(X, Y, Z) \implies families(X, Y) \end{array}$$

In Example 3.1, the arguments of mum now contain more information than the last one. For instance, in $mum(lily, victor, birth)$, the motherhood relation between two individuals conveys more information than knowing only that Lily gave birth to someone, or Victor has a birth mother. However, a repair system which, unlike a human agent, does not have unbounded commonsense background knowledge, will not be able to rank the informational value of the arguments this way. Some additional mechanism, such as the one we propose below, is required to track this difference.

The new argument splits each instance of the logical concept mum into different types. Some rules from $\mathbb{T}_m$ only apply to birth mothers in $\mathbb{T}_{rm}$ while other rules still apply to all types of mothers.

Increasing the arity of a predicate may seem equivalent to removing the predicate, and then adding that predicate with new arity. But that is not true, because the conceptual contraction causes informational loss, which the conceptual expansion function cannot then recover. To illustrate why

conceptual revision does not equal conceptual contraction followed by conceptual expansion, we can model the process of revising $\mathbb{T}_m$ to $\mathbb{T}_{rm}$ by assuming that a conceptual contraction function deletes all axioms of $mum/2$ from the input theory. Then all axioms of that predicate are lost. Even if that function 'backs up' the axioms being deleted, they cannot be directly used as the input for conceptual expansion, because they will now have incorrect arity. A new argument must be added, by an argument supplement function, to all the 'backed up' axioms to create new axioms with the appropriate arity for the conceptual expansion function. As a result, an argument supplement function is necessary, i.e. revision is the combination of contraction and expansion with the addition of an argument supplement function.

Like conceptual knowledge in a human mind, the components of a logical concept can thus become dynamic. When an element is redundant for a concept, the corresponding argument can be abandoned. Thus the arity of the corresponding predicate decreases. For example, to distinguish British citizenship and British residence, the relevant information about David is represented as $\Longrightarrow country(david, uk, citizen)$ and $\Longrightarrow country(david, uk, resident)$. If that policy changed, and all residents became citizens, the last argument would be redundant. So the axiom should be rewritten as $\Longrightarrow country(david, uk)$. Consequently, the arity of the predicate should decrease in line with the deletion of arguments that described a logical concept's redundant or obsolete features.

So far, we defined conceptual change, implemented using ABC's Reformation algorithm. This definition provides the theoretical base to our main topic: to measure signature entrenchment for ABC's Reformation repairs, as introduced in the next section.

4. Measuring Entrenchment

Conceptual change edits the signature of logical theories (Bundy & Mitrovic, 2016). Measuring the entrenchment of signature elements helps rank alternative repairs that effect conceptual changes, so that preferred repairs can be automatically prioritised.

Signature elements can be related to each other, depending on whether they are involved in a chain of rules. For instance, $fly(tweety)$ is a theorem in a faulty Datalog theory given in Example 4.1. $\mathbb{T}_b$ is incompatible because of its theorem $fly(tweety)$: $tweety$ is a penguin in $\mathbb{T}_b$, a flightless bird, but the theory implies $tweety$ can fly. If the predicate $bird$ is split, and the predicate in A1 is renamed to $birdFlying$, then $fly(tweety)$ will no longer be a theorem, and no longer conflict with the preferred structure. On the other hand, if the argument in $penguin(tweety)$ is renamed from $tweety$ to $liza$, the theorem $fly(tweety)$ will become $fly(liza)$ as a consequence. Instead of $tweety$, $liza$ will become the bird that can fly like $jonathan$, which resolves the conflict differently.

Example 4.1. *Bird Theory* $\mathbb{T}_b$.

$$
\begin{aligned}
bird(X) &\implies fly(X) && \text{(A1)}\\
bird(X) &\implies feathered(X) && \text{(A2)}\\
bird(X) &\implies wings(X) && \text{(A3)}\\
penguin(X) &\implies bird(X) && \text{(A4)}\\
seagull(X) &\implies bird(X) && \text{(A5)}\\
&\implies penguin(tweety) && \text{(A6)}\\
&\implies seagull(jonathan) && \text{(A7)}
\end{aligned}
$$

$\mathcal{T}(\mathbb{PS}_b) = \{penguin(tweety), feathered(tweety), seagull(jonathan), feathered(jonathan), fly(jonathan)\}$
$\mathcal{F}(\mathbb{PS}_b) = \{fly(tweety)\}$

It can be seen that when a predicate is involved in a rule, a change in the rule could result in corresponding changes in the predicates which occur in the same rules - especially those on the other side of the implication in that rule. Therefore, rules play an essential role in analysing and modifying a signature. As with the representation of relations between concepts for causal inference (cf. Sloman (2005) a *directed graph* can trace the inferential links between predicates in a theory.

Definition 4.1 (Theory Graph)**.** *A theory graph represents the links between predicates in a logical theory by a finite set of nodes and edges, where each predicate occurs exactly once.*

Node: *each node corresponds to a predicate together with its arity.*

Edge: *an arrow that connects to itself or to another node in the graph. The components of an edge include:*
Direction: *from a node in the body of the rule to the head of that rule;*
Label: *a 3-tuple including the name of the axiom and the arguments of the body node followed by the arguments of the head node.*

Path: *a path consists of the nodes connected by a sequence of edges in same direction.*

Given a theory graph, one can fully recovery its theory. The theory graph of rule $A1$ in equation (2), which has n propositions in its body and one proposition in its head, could be drawn as Figure 2. Each tail node corresponds to a proposition in the body of the rule and the head node represents the proposition in the head of the rule. An assertion can be seen as a rule without preconditions. In a theory graph, an assertion is drawn as a head node pointed by an edge with a special tail node of 'true' and the edge of a goal axiom, which has no head, has a special head node of 'false' e.g., A7 in Figure 3 and A1 in Figure 5 below, respectively.

$$A1.\ \bigwedge_{i=1}^{n} p_i(t_1^i, ..., t_m^i) \implies q(u_1, ..., u_k) \tag{2}$$

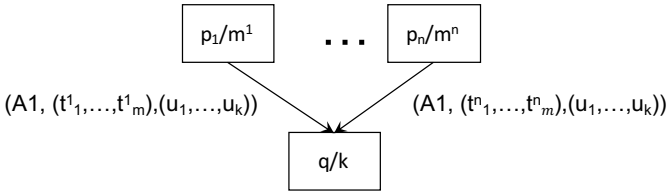

Figure 2. Theory sub-graph of A1 in Equation (2).

Definition 4.1 requires that no duplicates are allowed in a theory graph. When a predicate is involved in multiple axioms, there will be multiple edges corresponding to these axioms. In each edge, the first element of its label, which is the name of the axiom e.g., A1 in Figure 2, shows to which axiom the edge belongs. Hence different axioms can be represented correspondingly. When a predicate appears n times with different arguments in the body of a rule, there will be n edges from the predicate to that rule's head. Those edges will be different due to the distinct arguments. On the other hand, if a predicate appears in both the body and the head of a rule, then the labelled edge should go from the predicate to itself.

Theorem 4.1. *If there is no path from predicate p to predicate q in the theory's theory graph, then assertions of p cannot contribute to any proof of an assertion of q.*

Theorem 4.1 determines whether adding a theorem of p impacts building any of the proofs of q's instances. We can now return to Example 4.1, a faulty Datalog-like logical theory with seven axioms, and give its theory graph in Figure 3. The 'true' node corresponds to assertions made about individuals.

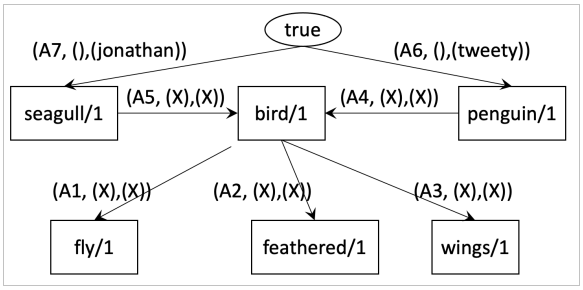

Figure 3. Theory graph of Example 4.1.

A theory graph clearly shows how predicates are linked inferentially. For example, in Figure 3, the predicate $wings$ is linked to $bird$ via A3. The change of a predicate could affect its linked ones, e.g., if $wings$ is changed into $teeth$, then the meaning of $bird$ in terms of its features is also altered.

The entrenchment of predicate symbols will be evaluated based on the defined theory graph, following which the entrenchment of an argument will be evaluated based on its argument domain.

4.1 Predicate Entrenchment

Given a $\mathbb{PS}$, predicates that occur in $\mathbb{PS}$, which we will call *protected predicates*, are most entrenched and fully trusted. How far a predicate p is linked to a protected predicate via the theory graph will correspond to how entrenched the predicate p is: the further p is to protected predicates, the smaller impact on the protected predicates p has when p is changed, and so the less entrenched (or easier to change) p must be.

As a result, to measure how entrenched a predicate is, we need to formalise how far the predicate is from protected predicates in the theory graph. To that end, we define the distance of a predicate p from the nearest protected predicate, using the term *confidence distance*.

Definition 4.2 (Confidence Distance)**.** *The confidence distance $pd(p)$ of a predicate p is the number of edges on the shortest path from p to its nearest protected predicate, following the direction of each arrow, or infinity if there is no such a path.*

$$pd(p) = \begin{cases} Min(\{|\vec{d}(p,q)| \ : \ \vec{d}(p,q) \in \mathcal{G} \wedge q \triangleleft \mathbb{PS}\}) \\ \infty, \ \nexists \vec{d}(p,q) \in \mathcal{G} \end{cases} \tag{3}$$

where $q \triangleleft \mathbb{PS}$ means that q is a predicate such that $\exists q(\vec{c}), \ q(\vec{c}) \in \mathcal{T}(\mathbb{PS}) \vee q(\vec{c}) \in \mathcal{F}(\mathbb{PS})$, $\vec{d}(p,q)$ is a path followed arrows between p and q in the theory graph $\mathcal{G}$.

Here a path means a set of nodes connected by edges following the direction from the tail to the head of an edge. When a predicate has no path to any protected predicates, it is called an *isolated predicate*. Following Definition 4.2, the confidence distance of an isolated predicate is infinite, which ensures that isolated predicates are always the least preferred. Accordingly, following equation (3), a protected predicate's confidence distance is the minimum value.

Our motivation is that the further a predicate is from a protected predicate pp, the more problems could happen in terms of proving the statements of pp. On the other hand, the nearer a predicate is from pp, the more important it is for the statements that prove pp. In other words, the nearer a predicate p is to protected predicates, the more impact it has on the proofs of $\mathbb{PS}$ when changing p.

The predicates occurring in the $\mathbb{PS}$ are fully trusted to be accurate by definition: the $\mathbb{PS}$ contains the most accurate label for each concept as predicate symbols, and the essential components of that concept as their arguments[3]. We assume that if the head of a rule has an accurate representation, then its body is represented accurately to some extent. Thus, a predicate nearer to ones in $\mathbb{PS}$ will be more accurate in its representation, because it is connected to a fully accurate representation through fewer rules.

Accordingly, Definition 4.3 gives a measure of the predicate entrenchment $e(p)$ for predicate p based on its confidence distance.

Definition 4.3 (Predicate Entrenchment)**.** *The entrenchment of a non-isolated predicate p_1 is given in terms of its confidence distance and the maximum confidence distance of all non-isolated predicates in the theory graph:*

3. These can be considered prototypes of that category, or observations of its exemplars. For our initial implementation we consider them reliable by definition. We leave cases where they are themselves subject to revision for future work.

$$e(p) = \begin{cases} 1 - \frac{pd(p)}{pd_{Max}+1}, \ pd(p) \neq \infty \\ \frac{1}{pd_{Max}+2}, \ pd(p) = \infty \end{cases} \tag{4}$$

where $pd_{Max} = \max(\{pd(q) : pd(q) \neq \infty\})$.

There are two disjoint kinds of predicates including, non-isolated predicate(p_1) and isolated predicate (p_2). A protected predicate p is always a non-isolated predicate. The defined measurement has desired properties as follows[4].

Theorem 4.2. *Based on the confidence distance, the important properties that predicate entrenchment* $e(p)$ *has are as follows, where* p, p_1 *and* p_2 *are predicates, and* $\mathbb{S}_p$ *is the set of predicates which occur in* $\mathbb{PS}$ *while* $\mathbb{S}_t$ *is the set of predicates which occur in the theory but not in* $\mathbb{PS}$.

1. $\forall p \in (\mathbb{S}_t \cup \mathbb{S}_p).\ e(p)$ *has exactly one value.*
   The entrenchment of a predicate should be just one value.

2. $\forall p \in (\mathbb{S}_t \cup \mathbb{S}_p).\ 0 < e(p) \leq 1$.
   The range of an entrenchment should be (0,1], where 0 means that a predicate is not trusted at all and 1 represents that the predicate is most entrenched and fully trusted.

3. $\forall p_2 \in \mathbb{S}_p.\ e(p_2) = 1 \ \wedge \ \forall p_1 \in \mathbb{S}_t.\ 0 < e(p_1) < 1$.
   Because $\mathbb{PS}$ is more trusted than the theory, a protected predicate is most entrenched so the entrenchment is 1. Any predicate appearing only in the theory is believed to some extent, but less than a protected predicate. Meanwhile, any predicate that occurs in the theory is considered to convey some information. Therefore, its entrenchment is bigger than 0 but smaller than 1.

4. $\forall p_1, p_2 \in \mathbb{S}_t.\ e(p_1) > e(p_2)$, *iff* $pd(p_1) < pd(p_2)$.
   When neither predicate occurs in $\mathbb{PS}$, p_1 is more entrenched than p_2 if and only if p_1 is closer to protected predicates in terms of its confidence distance. The smaller $pd(p_1)$ is, the more impact on $\mathbb{PS}$ changing p_1 will have.

Figure 4 shows the entrenchment of each predicate in $\mathbb{T}_r$ from Example 4.1. As $bird/1$ is the least entrenched predicate involved in the unwanted proof of $fly(tweety)$ using (A1, A4, A6), the repaired theory $\mathbb{T}_{rp}$ is prioritised: $birdFlying/1$ is split off from $bird/1$ as shown in Example 4.2.

The issue with $\mathbb{T}_r$ is that a logical concept of bird given in terms of flying ability is ambiguous: some, but not all birds can fly. Thus, $\mathbb{T}_{rp}$ is desired because it separately represents those flying birds by the new predicate $birdFlying$, and then axiom A5' is added to classify seagulls as flying birds. This is an example where a preferred repair is prioritised based on predicate entrenchment.

4. Proofs of theorems in this paper are available on GitHub: https://github.com/XuerLi/Publications/tree/main/ACS2021.

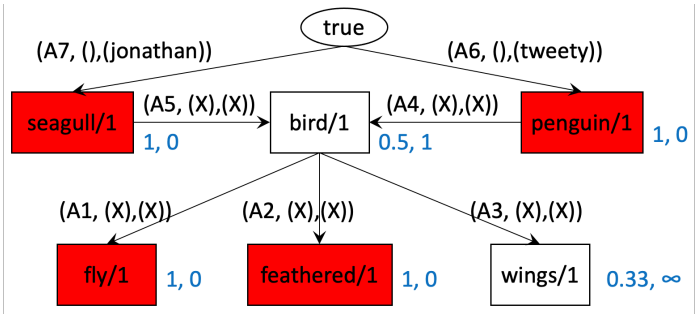

Figure 4. Theory graph of Example 4.1, with protected predicates highlighted in red, and entrenchment of each predicate X (and corresponding confidence distance Y, given in the form of X, Y) highlighted in blue.

Example 4.2. *A Preferred Repair ($\mathbb{T}_{rp}$) for the Faulty Bird Theory in Example 4.1.*

$$
\begin{aligned}
birdFlying(X) &\implies fly(X) && \text{(A1)}\\
bird(X) &\implies feathered(X) && \text{(A2)}\\
bird(X) &\implies wings(X) && \text{(A3)}\\
penguin(X) &\implies bird(X) && \text{(A4)}\\
seagull(X) &\implies bird(X) && \text{(A5)}\\
seagull(X) &\implies birdFlying(X) && \text{(A5')}\\
&\implies penguin(tweety) && \text{(A6)}\\
&\implies seagull(jonathan) && \text{(A7)}
\end{aligned}
$$

$\mathcal{T}(\mathbb{PS}_b) = \{penguin(tweety), feathered(tweety), seagull(jonathan), feathered(jonathan), fly(jonathan)\}$
$\mathcal{F}(\mathbb{PS}_b) = \{fly(tweety)\}$

In comparison, a dispreferred repair is given in Example 4.3. Here, a new predicate $flyAbnormal/1$ is split off from $fly/1$. It makes the theory correct w.r.t. $\mathbb{PS}$ but incorrect from an intuitive human perspective. A penguin cannot fly in any sort of sense, so *fly* is the wrong concept to change. And just as desired, predicate entrenchment does not prioritise this repair in favour of $\mathbb{T}_{rp}$.

Example 4.3. *A Dispreferred Repaired Bird Theory* $\mathbb{T}_{brb}$.

$$bird(X) \implies flyAbnormal(X) \quad \text{(A1)}$$
$$bird(X) \implies feathered(X) \quad \text{(A2)}$$
$$bird(X) \implies wings(X) \quad \text{(A3)}$$
$$penguin(X) \implies bird(X) \quad \text{(A4)}$$
$$seagull(X) \implies bird(X) \quad \text{(A5)}$$
$$seagull(X) \implies fly(X) \quad \text{(A5')}$$
$$\implies penguin(tweety) \quad \text{(A6)}$$
$$\implies seagull(jonathan) \quad \text{(A7)}$$

$\mathcal{T}(\mathbb{PS}_b) = \{penguin(tweety), feathered(tweety), seagull(jonathan), feathered(jonathan), fly(jonathan)\}$
$\mathcal{F}(\mathbb{PS}_b) = \{fly(tweety)\}$

4.2 Argument Entrenchment

Argument entrenchment is a measure of the contribution of arguments for a predicate. It is defined based on the diversity of the individuals that the argument in this predicate can represent. We have previously defined an argument's *domain* in Definition 3.1. The size of this domain reflects the diversity of the potential constants that can appear in place of that argument.

Example 3.1 Enriched Motherhood Theory $\mathbb{T}_{rm}$ with Conceptual Changes.

$$\implies mum(lily, victor, birth)$$
$$\implies mum(anna, david, birth)$$
$$\implies mum(anna, victor, step)$$
$$\implies mum(lucy, tom, birth)$$
$$mum(X, Y, birth) \wedge mum(Z, Y, birth) \implies X = Z$$
$$mum(X, Y, Z) \implies families(X, Y)$$

Recall $\mathbb{T}_{rm}$ in Example 3.1. The predicate mum has 3 arguments. The domains of mum's three arguments are $\mathcal{D}(mum, 1, \mathbb{T}_{rm}) = \{lily, anna, lucy\}$, $\mathcal{D}(mum, 2, \mathbb{T}_{rm}) = \{victor, david, tom\}$ and $\mathcal{D}(mum, 3, \mathbb{T}_{rm}) = \{birth, step\}$, respectively. Given the argument domains, without other information[5], the probability for an artificial agent to recover theorems that 'lost' one argument is the reciprocal of the size of the missed argument. For example, if the predicate $mum/3$ loses its first argument in $\mathbb{T}_{rm}$, e.g., $mum(\_, tom, birth)$, the probability of recovery of $mum(lucy, tom, birth)$ is the probability of randomly selecting $lucy$ from its domain $\mathcal{D}(mum, 1, \mathbb{T}_{rm})$, which is $1/3$. Similarly, it is $1/3$ for theorems that lose the second argument, e.g., $mum(lucy, \_, birth)$, and $1/2$ for ones that lose the third argument, e.g., $mum(lucy, tom, \_)$.

5. For example, who might live in the same city, or who might have the same address.

In Equation (5), the function $\sigma(X, Y, Z)$ returns the probability of recovering the theorem[6] X which loses its Yth argument from the original theory Z.

$$\sigma(mum(lucy, tom, birth), 1, \mathbb{T}_{rm})) = \frac{1}{|\mathcal{D}(mum, 1, \mathbb{T}_{rm})|} \tag{5}$$

We assume that *the larger the recovery probability is, the less informational value that argument has*. As entrenchment formally captures the informational value of an element, the entrenchment of an argument can be evaluated based on its domain size.

Definition 4.4 (Argument Entrenchment)**.** *Let $\mathcal{E}_a(p, i, \mathbb{T})$ be the entrenchment of the i^{th} argument of predicate p in theory $\mathbb{T}$:*

$$\mathcal{E}_a(p, i, \mathbb{T}) = \begin{cases} |\mathcal{D}(p, i, \mathbb{T})|, \ p \not\lhd \mathbb{PS} \\ \infty, \ p \lhd \mathbb{PS}, \end{cases} \tag{6}$$

where $1 \leq i \leq arity(p)$.

Because $\mathbb{PS}$ is the benchmark of the correctness of both the theory and its signature. The argument entrenchment of an argument involved in $\mathbb{PS}$ is bigger than the others.

Theorem 4.3. *The argument entrenchment of an argument that occurs in $\mathbb{PS}$ is bigger than one that only occurs in the theory, and not in $\mathbb{PS}$.*

$$\forall p \lhd \mathbb{PS}, p' \not\lhd \mathbb{PS}, \ \mathcal{E}_a(p, i, \mathbb{T}) > \mathcal{E}_a(p', j, \mathbb{T}) \tag{7}$$

where $1 \leq i \leq arity(p)$ and $1 \leq j \leq arity(p')$.

When multiple repairs change the arguments of a predicate, the ones which cause the smallest argument entrenchment scores' reduction are ranked at the top, e.g., Example 5.4 in §5.

In Example 3.1, the third argument of $mum/3$ is least entrenched. Its deletion is prioritised compared to deleting the other two arguments: e.g. a repair insensitive to argument entrenchment might delete the first argument, while a repair based on argument entrenchment would chose the third. Knowing which individual is someone's mother contains more information than only knowing someone has one type of mother vs. another. We compare the entrenchment based on the repaired theories, so the entrenchment of newly added arguments is also calculated.

5. Evaluation

Our hypothesis, as stated in Definition 2.6, is that ranking repairs using signature repair operations based on signature entrenchment scores will prioritise preferred repairs: repairs that are consistent with the $\mathbb{PS}$, parsimonious, and intuitive. The repairs that change the least entrenched predicates, or cause the smallest deduction on argument entrenchment scores, should thus match preferred repairs. *Our evaluation thus compares the entrenchment scores of preferred repairs and dispreferred repairs*.

6. In ABC, proving ground propositions such as theorems is sufficient, when checking for faults based on $\mathbb{PS}$, because $\mathbb{PS}$ itself only consists of ground propositions.

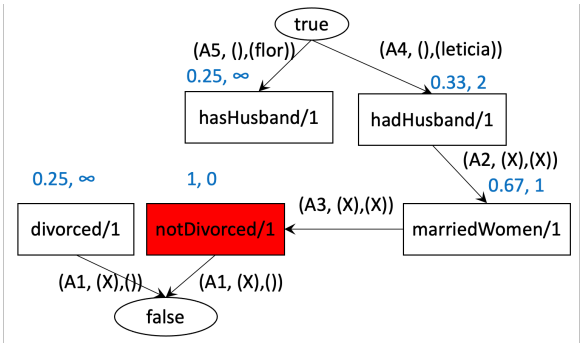

Figure 5. Theory graph of Example 5.1 with protected predicates highlighted in red, and entrenchment of each predicate X, and the corresponding confidence distance Y (given in the form X, Y) highlighted in blue.

The theories evaluated in ABC are adapted from the knowledge representation literature - they are not, at this stage, derived from human behaviour - and their $\mathbb{PS}$s are formalised to represent explicit statements or desired theorems from that literature. As we focus on the analysis of signature entrenchment for conceptual changes, only theories that require conceptual changes are discussed.

Example 5.1 gives the faulty $Married\ Women$ theory, whose theory graph is shown above in Figure 5. The proof of the incompatibility of $notDivorced(leticia)$ uses $(A4, A2, A3)$. Among all the predicates involved in the proof, $hadHusband$, with score 0.33 is the least entrenched, compared with $marriedWomen$ with 0.67 and $notDivorced$ with 1. Thus, the most preferred repair according to entrenchment is replacing the prior instance of predicate $hadHusband$ in $A2$ with $hasHusband$.

Example 5.1. *Married Women Theory.*

$$
\begin{aligned}
divorced(X) \wedge notDivorced(X) &\implies && \text{(A1)}\\
hadHusband(X) &\implies marriedWoman(X) && \text{(A2)}\\
marriedWoman(X) &\implies notDivorced(X) && \text{(A3)}\\
&\implies hadHusband(leticia) && \text{(A4)}\\
&\implies hasHusband(flor) && \text{(A5)}
\end{aligned}
$$

$\mathcal{T}(\mathbb{PS}) = \{notDivorced(flor)\}, \mathcal{F}(\mathbb{PS}) = \{notDivorced(leticia)\}$

Example 5.2. *The Preferred Repair of the Married Women Theory.*

$$\begin{aligned} divorced(X) \wedge notDivorced(X) &\implies && \text{(A1)}\\ hasHusband(X) &\implies marriedWoman(X) && \text{(A2)}\\ marriedWoman(X) &\implies notDivorced(X) && \text{(A3)}\\ &\implies hadHusband(leticia) && \text{(A4)}\\ &\implies hasHusband(flor) && \text{(A5)} \end{aligned}$$

$\mathcal{T}(\mathbb{PS}) = \{notDivorced(flor)\}, \mathcal{F}(\mathbb{PS}) = \{notDivorced(leticia)\}$

Example 5.3 gives a dispreferred repair, which renames $marriedWomen/1$ in A3 with $hasHusband/1$. Although the theory respects its $\mathbb{PS}$, it violates the intuitive judgement that if a woman once had a husband, that does not mean she is necessarily still married. We thus want to automatically rank this repair lower than alternatives.

Example 5.3. *A Dispreferred Repair of the Married Women Theory.*

$$\begin{aligned} divorced(X) \wedge notDivorced(X) &\implies && \text{(A1)}\\ hadHusband(X) &\implies marriedWoman(X) && \text{(A2)}\\ hasHusband(X) &\implies notDivorced(X) && \text{(A3)}\\ &\implies hadHusband(leticia) && \text{(A4)}\\ &\implies hasHusband(flor) && \text{(A5)} \end{aligned}$$

$\mathcal{T}(\mathbb{PS}) = \{notDivorced(flor)\}, \mathcal{F}(\mathbb{PS}) = \{notDivorced(leticia)\}$

In Example 5.4, the $Families$ theory, on the left side, is faulty because it does not consider step-parents to be part of a child's family. Its preferred repair is provided on the right side, where the constant $birth$ is extended into a variable in the rule. Thus, the fault is repaired in the way of retaining the concept of different kinds of parents. In contrast, Example 5.5 gives two dispreferred but potentially good repairs. The left one renames $step$ to $birth$, and the right one deletes the third argument of predicate $parent$. Based on human judgement, the repairs in Example 5.5 are not preferred because they either lose the concept of the step-parent, or abandon kinds of parent entirely. However they could be good repairs if the concept of different kinds of parent became redundant.

The argument entrenchment scores' deduction from these three repairs is 0, 1 and 2, respectively. Our entrenchment algorithm thus ranks the preferred repair at the top, as desired, then the left repair in Example 5.5. The right one in Example 5.5 is ranked last, because it loses the most information.

Example 5.4. *Faulty Families Theory and its Preferred Repair.*

$$\begin{array}{l|l} parent(X,Y,birth) \implies families(X,Y) & parent(X,Y,Z) \implies families(X,Y) \\ \qquad\qquad\qquad\quad \implies parent(a,b,birth) & \qquad\qquad\qquad\quad \implies parent(a,b,birth) \\ \qquad\qquad\qquad\quad \implies parent(a,c,step) & \qquad\qquad\qquad\quad \implies parent(a,c,step) \end{array}$$

$$\mathcal{T}(\mathbb{PS}) = \{families(a,b),\ families(a,c)\},\ \mathcal{F}(\mathbb{PS}) = \emptyset$$

Example 5.5. *Dispreferred Repairs of the Faulty Families Theory.*

$$\begin{array}{l|l} parent(X,Y,birth) \implies families(X,Y) & parent(X,Y) \implies families(X,Y) \\ \qquad\qquad\qquad\quad \implies parent(a,b,birth) & \qquad\qquad\quad \implies parent(a,b) \\ \qquad\qquad\qquad\quad \implies parent(a,c,birth) & \qquad\qquad\quad \implies parent(a,c) \end{array}$$

$$\mathcal{T}(\mathbb{PS}) = \{families(a,b),\ families(a,c)\},\ \mathcal{F}(\mathbb{PS}) = \emptyset$$

6. Conclusion

In this paper, we described a novel way to rank the automated repair of logical theories that represent conceptual knowledge, through measuring the entrenchment of elements in those theories' logical signatures. As with human concepts, some logical concepts are more valuable than others - reflected in our method by ranking concepts according to their links to a fixed store of reliable observations, the preferred structure. This ranking enables the ABC system to prioritise those signature repairs to a faulty Datalog theory that cause the smallest reduction in entrenchment scores. Our algorithm only requires reference to the preferred structure: one set of true propositions and one of false ones. Because the preferred structure was already in use by ABC to detect and repair faults, no additional information is required to define and implement signature entrenchment.

This novel method for theory repair remains restricted in several ways: 1) It is currently limited to decidable theories. Though it could be applied to non-decidable theories, heuristics would be essential to limit the total number of theorems. 2) It cannot cross-evaluate predicate entrenchment and argument entrenchment scores against each other, 3) It is not yet compatible with *probabilistic* concept representations, as often found in causal models of cognition (Sloman, 2005; Danks, 2014), despite our entrenchment theory graphs sharing some of the same surface structure; although ABC's computational complexity is less than that required by a probabilistic graph. 4) It has not yet been empirically validated by human judgements over concepts in the same or similar domains (such as birds, or some equivalent artificial animal category), or by a wider measure of human judgements of repair plausibility in a controlled setting. The further development of empirical measures for signature entrenchment, and human intuition on repairs overall, is thus our most pressing next step.

References

Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, *50*, 510–530.

Bundy, A. (1983). *The computational modelling of mathematical reasoning*. Academic Press.

Bundy, A., & Mitrovic, B. (2016). *Reformation: A domain-independent algorithm for theory repair*. Technical report, University of Edinburgh.

Danks, D. (2014). *Unifying the mind: Cognitive representations as graphical models*. MIT Press.

Frankfurt, H. (1958). Peirce's notion of abduction. *The Journal of Philosophy*, *55*, 593–597.

Gärdenfors, P. (1988). *Knowledge in flux: Modeling the dynamics of epistemic states.*. MIT press.

Gärdenfors, P. (2003). *Belief revision*. Cambridge University Press.

Gopnik, A., & Wellman, H. M. (1992). Why the child's theory of mind really is a theory. *Mind & Language*, *7*, 145–171.

Kowalski, R. A., & Kuehner, D. (1971). Linear resolution with selection function. *Artificial Intelligence*, *2*, 227–60.

Li, X. (2021). *Automating the repair of faulty logical theories*. Doctoral dissertation, School of Informatics, University of Edinburgh.

Li, X., Bundy, A., & Smaill, A. (2018). ABC repair system for Datalog-like theories. *KEOD* (pp. 333–340).

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*, 207–238.

Murphy, G., & Brownell, H. (1985). Category differentiation in object recognition: typicality constraints on the basic category advantage. *Journal of experimental psychology. Learning, memory, and cognition*, *11 1*, 70–84.

Pickering, M. J., & Garrod, S. (2013). An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*, *36*, 329–347.

Posner, M. I., & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology*, *77*, 353–363.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, *8*, 382–439.

Schmid, H.-J. (Ed.). (2016). *Entrenchment and the psychology of language learning: How we reorganize and adapt linguistic knowledge*. De Gruyter Mouton.

Sloman, S. (2005). *Causal models : How people think about the world and its alternatives*. Oxford University Press, USA.

Urbonas, M., Bundy, A., Casanova, J., & Li, X. (2020). The use of max-sat for optimal choice of automated theory repairs. *Artificial Intelligence XXXVII* (pp. 49–63). Cham: Springer.