Advances in Cognitive Systems 8 (2019) 93–112 Submitted 5/2019; published 12/2019

Heuristic Construction of Explanations Through Associative Abduction

Pat Langley PATRICK.W.LANGLEY@GMAIL.COM
Institute for the Study of Learning and Expertise, Palo Alto, California 94306 USA
School of Computer Science, University of Auckland, Private Bag 92019, Auckland 1142 NZ

Ben Meadows BMEA011@AUCKLANDUNI.AC.NZ
Department of Electrical and Computer Engineering, University of Auckland, Auckland 1142 NZ

Abstract

Humans regularly explain observations of their environment in terms of background knowledge. This process is better characterized as abduction than as deduction, since it often requires introduction of assumptions about unobserved relations. In this paper, we present a theory of abductive explanation that builds on earlier work but extends it in new directions. The theory distinguishes between definitions and constraints, with the former used to elaborate existing explanations and the latter used to detect and repair inconsistencies. We also describe PENUMBRA, an implemented system that instantiates the theory, and demonstrate its behavior on a number of domains. The system carries out heuristic search through a space of explanations, processing observations incrementally, and generating alternative accounts for a given set of inputs. We conclude by discussing related approaches to explanation, limitations of the implementation, and directions for future research.

1. Background and Motivation

One distinctive feature of human cognition is the ability to understand complex situations and events. This invariably involves explaining observations in terms of available knowledge. Moreover, these explanations are typically *abductive* in character, in that they incorporate plausible assumptions that are neither observed nor derived deductively. Abductive explanation is a general ability that arises in many contexts, from sentence processing and story understanding (Winston, 2012) to scene interpretation and plan recognition (Blaylock & Allen, 2005) to diagnosis (Reggia, Nau, & Wang, 1985). We would like a computational theory of the structures that underlie this capacity and the processes that operate over them. Ultimately, this should contribute to a more comprehensive cognitive architecture (Langley, Laird, & Rogers, 2009) that supports goal-directed activity over time, but here we will focus only on conceptual understanding.

Let us consider a simple example. Suppose we are told that Abe possesses some cash and Bob possesses a car, but that later Abe possesses the same car. Although we did not observe any transaction, we can reasonably assume that one took place. Two explanations come immediately to mind. One is that Abe bought the car from Bob using money; another is that Abe stole the car from Bob by threatening him in some way. We also know these two explanations are mutually

exclusive, in that purchases and robbery are two distinct ways to transfer possession of objects. This means that we must not only introduce plausible assumptions about unobserved events, but consider the competing explanations and keep them separate. Later, we may hear that Abe actually gave money to Bob, eliminating theft as an alternative. More complex examples would involve multi-step inference chains that generate hierarchical accounts of observations.

In this paper, we present a cognitive systems account of such abductive explanations. Our analysis draws on standard ideas from the paradigm, including a focus on high-level cognition, the importance of structured representations and knowledge, a reliance on heuristic search, and incorporation of constraints from human behavior, such as incremental processing of observations. The approach that we describe builds directly on two earlier efforts (Bridewell & Langley, 2011; Meadows, Langley, & Emery, 2014) in this area, but extends them to incorporate richer representations and novel reasoning mechanisms. In the next section, we discuss two formulations of the abductive explanation task, along with prior results in each framework, and clarify our reasons for selecting one of them. After this, we describe a new theory for this ability, focusing first on assumptions about cognitive structures and then on processes that inspect and manipulate them. Next we report PENUMBRA, an implemented system that instantiates these theoretical ideas, along with its behavior on multiple scenarios that demonstrate its coverage. We close by discussing links to earlier work, noting limits of the implementation, and proposing directions for additional research in this area.

2. Two Formulations of Abductive Explanation

We can define any cognitive task in terms of the information provided as inputs and the content generated as outputs. However, there are often different ways to translate an informal problem into a formal specification. The literature on abductive explanation has explored two distinct statements of this mental task that we should discuss before proceeding further. These treatments are orthogonal to whether inputs are processed incrementally, an important feature of human processing whose discussion we will delay until a later section.

The first formulation borrows from classic treatments of abduction in logic and the philosophy of science (Peirce, 1878; Hempel, 1966). We can state it as:

- *Given*: A set of general knowledge elements K (e.g., relational rules)
- *Given*: A set of specific observed facts O (e.g., relational literals)
- *Find*: A set of specific plausible assumptions A (e.g., relational literals)
- *Find*: A set of proof trees P that derive elements of O from A and other elements of O with K

The key idea here is that the resulting explanation, a set of linked proof trees, must contain a proof for each observed fact. These may include default assumptions as terminal nodes, which can be shared across different proof trees, but each observation must follow deductively from these assumptions and from other observations by reasoning over available knowledge. Proof trees may correspond to causal chains, as in many scientific explanations, but this is not a requirement.

We will refer to this formulation as *derivational abduction* because observations must be derived from other beliefs. This paradigm has received considerable attention in the AI community. For example, Reggia et al. (1985) adopted the approach in their work on diagnosis, which inferred unobserved diseases that caused observed symptoms, and Hobbs et al. (1993) used it in their ap-

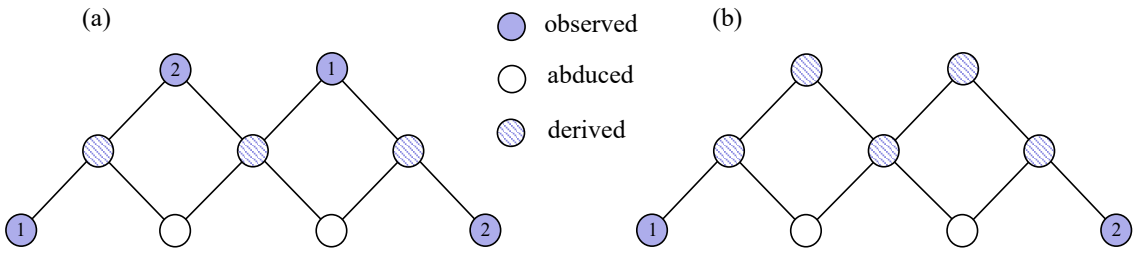

Figure 1. Two formulations for abductive explanation: (a) in derivational abduction, observations are the roots of proof trees but can also appear as terminal nodes; (b) in associative abduction, observations appear only as terminal nodes along with plausible assumptions.

proach to language processing, which posited logical relations that let it derive sentences' meanings. Ng and Mooney (1990) invoked the same scheme to address high-level aspects of story understanding, while using different criteria for evaluating candidates. More recent work in this tradition comes from Molineaux et al. (2012), Eckroth and Josephson (2014), Friedman et al. (2018), and Gordon (2018). The latter has applied an incremental variant of the framework to infer complex explanations of extended interactions among multiple agents. Derivational abduction is arguably the default approach taken by those interested in the topic. Figure 1 (a) depicts the structure of an explanation in this framework. Each observation appears as the root node for a proof tree in a connected set of such trees, although it may also serve as a terminal node in the proof for another observed element. Assumptions always appear as terminal nodes in one or more proof trees, often supporting more than one observation. The figure depicts these dependency relationships without specifying node contents, but these are typically atomic expressions in predicate or propositional logic.

The second formulation of abductive explanation shares many features with the classic treatment, but also differs in an important way. We can state it as:

- *Given*: A set of general knowledge elements K (e.g., relational rules)
- *Given*: A set of specific observed facts O (e.g., relational literals)
- *Find*: A set of specific plausible assumptions A (e.g., relational literals)
- *Find*: A set of proof trees P that derive new beliefs from O and A with K

The core distinction here is that observed facts appear only as terminal nodes in the proof trees, along with default assumptions. Other beliefs follow from them deductively, but there is no need to derive the observations themselves. This does not rule out the possibility of observed elements appearing as nonterminal nodes, but this is neither typical nor required. The sense of explanation here differs from that used in science, and it seldom has a causal interpretation, coming closer to 'making sense' of everyday situations, events, or activities.

We will refer to this framework as *associative abduction* because it treats observations as explained if they stand together, as in 'guilt by association'. This idea has been explored in the AI literature, but it is less common than the alternative. For instance, work on truth maintenance systems (de Kleer, 1986) takes this view, and Makatchev, Jordan, and VanLehn (2004) used it to

analyze students' defenses of answers to physics problems. Bridewell and Langley (2011) adopted the formulation in their AbRA system, which they presented as an account of everyday reasoning in humans. More recently, Meadows, Langley, and Emery (2013, 2014) reported a similar approach in their UMBRA system for plan understanding, which Langley et al. (2014) applied to dialogue interpretation. We will build directly on this latter work, which means that we will adopt the same formulation. Figure 1 (b) shows the structure of an explanation in this second paradigm. Here each observation appears as a terminal node in the set of connected proof trees, with assumptions at the same level. Nonterminal nodes are derived from the nodes below them in a given proof tree. Observations, assumptions, and nonterminals may appear in more than one derivation chain and thus support multiple inferences. Again, the figure is abstract, but nodes usually correspond to ground atomic expressions in some logical notation, possibly with unbound skolem symbols as arguments.

We should also mention other paradigms that have addressed abductive explanation. Early research on story understanding (e.g., Schank & Abelson, 1977) relied on 'scripts' to interpret action sequences, which comes closest to the second formulation in its emphasis on reasoning about everyday experience. The same holds for structural analogy (e.g., Gentner & Forbus, 1991), which retrieves relational cases from memory to interpret new situations and generate abductive inferences about them. Bayesian networks (Darwiche, 2009) support inference about unobserved variables from observed ones, but they can adopt either formulation depending on their structure, although relational variants (e.g., Charniak & Shimony, 1990; Kate & Mooney, 2009; Raghavan & Mooney, 2010) usually assume the first formulation of abduction. Finally, answer set programming (Baral, 2003) operates over logical rules to generate different possible 'worlds' that make distinct assumptions. We can view each world as explaining observed facts, but these do not involve proof trees, so they do not map well onto either the derivational or associative frameworks.

3. A Theory of Associative Abduction

Now we can turn to our theory of associative abduction, which is an instance of the second formulation. Any theory must address a set of phenomena that it aims to explain. In this case, we can identify five primary abilities that humans exhibit when they attempt to explain situations and events. These include the capacity to:

- Explain observations by connecting them to each other through available knowledge;
- Introduce plausible assumptions about relations or events that are not directly observed;
- Process observations incrementally on arrival and incorporate them into existing explanations;
- Detect and address conflicting beliefs that keep explanations from being consistent; and
- Generate alternative explanations when there are multiple plausible accounts of the observations.

We desire a computational account of these five cognitive functions. Previous work in the associative abduction framework on AbRA (Bridewell & Langley, 2011) and UMBRA (Meadows et al., 2014) has addressed the first three abilities. They also dealt with consistency by avoiding inferences that led immediately to conflicting beliefs, but lacked mechanisms for repairing such problems. Moreover, the programs carried out greedy search through the space of explanations and found only a single candidate even when other plausible accounts were possible. In this section, we propose an extended theory that builds on ideas in these efforts but that responds to all five target phenomena.

3.1 Representational Postulates

Before we can discuss our theory's claims about processes, we must specify its representational assumptions. Because explanation always relies on knowledge, we must first take a position on the types of knowledge in long-term memory and the forms they take.

R1. Explanatory inference draws on two types of knowledge structures: *definitions* and *constraints*.

This distinction is similar to the one that Simon (1970) has made in his analysis of scientific theories. Definitions are rules that specify higher-level relational predicates as conjunctions of more basic relations. These encode content similar to Prolog rules, in which the same predicate can appear as one or more heads. In contrast, constraints specify that two or more relations are mutually exclusive. These encode content similar to rules with empty heads in logic programming, although they specify two or more mutually exclusive relations. In our framework, such constraints may have heads, but these serve only to provide a common name to a set of alternatives.

For example, knowledge about English might include one definition for the type of noun phrase, NP1, that has an article followed by a noun, and another rule for a different type of noun phrase, NP2, that comprises an article, an adjectival phrase, and a noun. The same grammatical knowledge base might include a constraint that states an NP1 and an NP2 noun phrase cannot occur together if they involve the same noun token. Similarly, we might have knowledge that a given prepositional phrase can be a constituent for a noun phrase or a verb phrase, but not for both at the same time. Together, definitions and constraints define a space of possible combinations over the available predicates. This organization is similar to that found in many logic programs.

Now we can turn to the content and structure of explanations in our associative framework. The postulates here repeat some material from the previous section, but we include them for the sake of completeness. Our second assumption concerns the types of elements that appear in an account:

R2. Explanatory inference involves three types of beliefs: *observed*, *derived*, and *abduced*.

All beliefs take the same structural form. For example, they might be relational literals, that is, predicates with a set of arguments, such as *(on a b)*, or a frame instance, such as *(on ^top a ^bottom b)*. These arguments will often be constant terms, but others may be skolems, as in *(on ∗s1 c)*, to indicate an unidentified entity or event. In addition, each belief is marked as having a specified origin. Observed beliefs come from the external environment, derived beliefs have been deduced from other beliefs using knowledge, and abduced beliefs have been introduced as assumptions.

Of course, an explanation must involve more than a set of beliefs. It must also specify how knowledge links them to each other, which leads to another theoretical tenet:

R3. *Justifications* are instances of definitional rules that specify how derived beliefs follow from observed, abduced, and other derived beliefs.

Such justifications are similar to steps in a traditional proof tree, the difference being only that some of their antecedents may be abduced rather than observed from the environment or derived from other elements. These instantiated versions of definitions provide the connective tissue that allows the encoding of larger-scale entities.

We can combine these four types of elements into an explanatory structure that relates them. Our fourth representational postulate specifies this organization:

R4. An *explanation* is a connected proof graph that links observed, derived, and abduced beliefs by justifications, with derived beliefs as nonterminal nodes and others as terminal nodes.

We have already seen an abstract explanation of this variety in Figure 1 (b), with different colors denoting different types of beliefs.[1] In this framework, an explanation must be a single connected proof graph; this may have multiple root nodes, but every belief must be connected, directly or directly, to every other one. If two subsets of beliefs participate in two graphs with no shared elements, then we view them as distinct but not competing explanations.

However, we still require some way to group disconnected sets of beliefs that complement each other. This leads to yet another theoretical assumption:

R5. A *world* is a set of complementary beliefs, justifications, and associated explanations.

Because we cannot always expect an individual explanation to account for all observations, we need some type of structure – a world – that shelters them under a single umbrella.

Justifications in an explanation are instantiated versions of definitions that connect derived beliefs to others, but we have not mentioned constraints. These play a key role in our sixth postulate:

R6. An *active world* is a set of beliefs and associated explanations that is not known to violate any constraints; an *inactive world* is a set of beliefs that is known to violate constraints.

Note the qualifiers 'is not known' and 'is known' here. As we will see, one must detect inconsistencies before addressing them, and this requires cognitive attention. A set of beliefs may violate a constraint without it being realized, so whether a world is active depends on such detection.

Finally, recall that we are concerned not only with individual explanations for a set of observations, but with different accounts for the same content. Thus, we need a set of worlds that encode competing alternatives, with inconsistency being a key feature in their evaluation. This leads to another structural assumption, which concerns the organization of worlds:

R7. A *world history* is a phylogenetic tree that traces the creation of some worlds from others, with terminal nodes being active and nonterminals being inactive.

The root node of this history is the original world that contains observed beliefs and any inferences that have revealed no inconsistencies. Children are variations on their parents that address a single constraint violation, with terminal nodes denoting worlds in which no consistencies have yet been detected. We refer to this organization as a 'phylogenetic' tree because it is directly analogous to the structures in biology that specify the hypothesized evolution of organisms.

One conundrum raised by our assumption of multiple worlds is that, although humans have limited working memories, they can still keep a number of alternative hypotheses in mind. We maintain that the memory load is mitigated by the fact that many beliefs are shared across worlds, so that the reasoner must only keep track of ways in which these worlds differ. This suggests a final postulate about the cognitive structures that support explanatory inference:

1. An explanation need not include abduced beliefs; all terminal elements may be observed, as in typical parse trees.

R8. Beliefs are stored in a *single working memory*, with each element specifying the active worlds in which it *does not* hold.

According to this claim, the various worlds' beliefs are encoded in a distributed manner that takes advantage of shared observations, assumptions, and derivations. This scheme avoids the need for repeatedly making the same inferences during reasoning, as applying a rule can alter contents of multiple explanations, giving an implicit form of parallelism. Storing worlds for which a belief does not hold will reduce memory load when elements shared across worlds outnumber their differences. Thus, it serves as a heuristic measure that offers no guarantees but that should often be effective.

Most of these theoretical ideas have appeared in the literature on automated reasoning, logic programming, and abductive inference, which means that our representational postulates draw on a long and respected tradition. The notions of a world history that tracks the evolution of beliefs and the distributed encoding of worlds in a single working memory appear to be more novel. However, it is the combination of these elements into a unified framework that makes our structural account of abductive explanation an important intellectual contribution.

3.2 Processing Postulates

Now that we have presented our representational tenets, we can turn to processes that our theory posits operate over these data structures. We have already mentioned the first postulate, which links directly to characteristics of human explanation:

P1. Explanatory inference is an *incremental process* that updates worlds as it encounters new observations and draws new conclusions.

In other words, we assume that observed beliefs come from some external source one or a few at a time, which in turn drives inferential activity. For instance, when understanding a sentence, humans read only a few words, incorporate them into a partial parse tree or meaning structure, and then process later words. This has implications for cognitive processing, in that inferences made early on, with incomplete information, may need to be revised when more evidence becomes available. Connecting observed, derived, and abduced beliefs in explanations through justifications, as postulate **R4** states, should support extension and revision of worlds as new facts arrive.

Explanations can be complex structures that include many observed, abduced, and derived beliefs, along with justifications that connect them. In many situations, different explanations are possible for the same observations, and their construction requires that one consider these alternatives. This suggests a second postulate about processing:

P2. Explanatory inference involves *heuristic search through a space of possible worlds* that is defined by available knowledge and driven by observations.

Given that there may be competing accounts of the same facts, some form of search seems natural, but our theory also claims that *heuristic* search is involved. This constrasts sharply with approaches that rely on answer set programming (Baral, 2003), which generate all minimal consistent worlds and rank them afterwards. Rather, we assume that only promising worlds and their component explanations are pursued and extended, which is implicated in human processing.

Of course, we must still specify the operators that are responsible for generating explanations and traversing the space of worlds. Our third processing tenet refers to these mechanisms:

P3. The search for explanations relies on two types of operations: *elaboration* and *repair*.

These two processes complement each other, with the first one extending existing worlds by adding to their explanations and the second handling inconsistences that result from these elaborations. We will not take a position here on how these mechanism alternate or which one takes precedence, but we will claim that both are needed for a complete account of abductive explanation.

Let us consider the first activity in more detail, which occurs when one retrieves a definitional rule that unifies with one or more existing beliefs. More specifically, we postulate that:

P4. Elaboration involves the *application of a definitional rule* that produces a new derived belief, along with abduced beliefs for any unmatched antecedents.

The result is the creation of a derived belief based on the head of this instantiated rule. When all antecedents are matched, this inference step has a deductive character; however, if some antecedents are unmatched, then the process introduces assumptions for them that become abduced beliefs. For instance, given definitional rules for a context-free grammar, it would introduce constituents for noun phrases, verb phrases, and the like. Elaboration is purely monotonic, in that it only adds beliefs to a given world and it does not lead to new worlds.

The second activity comes into play when one notices that a world includes two or more inconsistent beliefs and is responsible for resolving the problem. In particular, we assume that:

P5. Repair involves *detecting* a violated constraint, *deactivating* the worlds it makes inconsistent, and *activating* new child worlds that are more nearly consistent.

Here the result is deactivation of the inconsistent world and creation of two or more children, each of which avoids the problem. This involves identifying beliefs that violate a constraint, finding the assumptions on which they depend, and creating disjoint sets, each of which eliminates the inconsistency. Each child world retains one of these sets, ensuring they sidestep the detected problem. For example, when processing a sentence, inferring two types of noun phrase for a given noun violates a constraint that only one may appear, leading to one world for each parse. Repair is nonomotonic, in that it removes abduced and derived beliefs from the child worlds, which then become active.

For a given set of worlds, or even for a single partial explanation, many definitional rules might be applied, and there may even be multiple inconsistences that should be resolved. This suggests the need for guidance, which leads to our sixth postulate about processing:

P6. Explanatory inference *focuses attention* on one belief at a time, which provides heuristic guidance to elaboration and repair by limiting which rules to consider.

The theory claims that the inference process never considers its entire knowledge base. Instead, it attends to a single belief at a time and only considers rules, both definitions and constraints, that connect with it. This means that the mechanism may overlook some useful elaborations or important inconsistencies, but, to the extent its heuristics for selecting foci are effective, such events will be rare. We might instead assume that the reasoner considers all possible matches at once, as in classic production systems (Klahr, Langley, & Neches, 1987), but organizing processing around foci has

two important effects. First, processing only considers rules that unify with the focus, which makes partial matching computationally tractable and scalable. Second, it provides a variety of 'spreading activation' that operates over generic rules rather than ground facts. That is, it produces a 'stream of consciousness' in which one idea leads to another, which is a key feature of human cognition.

As with our theory's structural postulates, these process elements draw on ideas explored by previous researchers, including work on automated reasoning, heuristic search, belief revision, and models of human cognition. The contribution of these elements lies not in isolation but in their integration into a coherent framework for conceptual inference and abductive explanation. As noted earlier, this should ultimately be embedded in a broader computational theory of the cognitive architecture that we will not address here. Such an architecture would use inferred beliefs, and the worlds associated with them, to make decisions during plan generation and execution, and thus drive an agent's behavior in an external environment.

4. The PENUMBRA Abduction System

We have implemented a system, PENUMBRA, that constitutes an instance of the theory described in the previous section.[2] Theories are by nature abstract and require additional assumptions to make them operational (Langley, 2018). In this section, we examine PENUMBRA's representational structures and processes, referring back to the earlier postulates when appropriate. Many implementation decisions are arbitrary and could arguably be done in other ways, but they are needed to produce the desired behavior, so we have done our best to document them here.

4.1 Representation in PENUMBRA

The PENUMBRA architecture incorporates most of the representational commitments that we enumerated earlier. To make these postulates concrete, the system adopts a specific syntax for encoding long-term and short-term cognitive structures. This offers a programming language for specifying domain knowledge and beliefs that underlie associative explanations. The notation has many similarities to those used in logic programming, even though the manner in which mental elements are processed differs in important ways.

For example, Table 1 (a) presents two definitional rules that describe ways a person can obtain an object from some other person who possesses it. These include definitions for buying the target object and for robbing its owner. Some elements, such as *(possess ?from ?obj ?t1 ?t2)*, describe situational relations, while others, such as *(give ?to ?from ?obj ?money ?t2 ?t3)* and *(threaten ?to ?from ?obj ?t2 ?t3)*, refer to activities. These relations include start and end times that specify intervals of time over which they hold, but PENUMBRA does not require them. Also, the implementation actually uses a frame-like notation that associates each argument with an attribute; this lets one omit arguments that are not needed in antecedents. Table 1 (b) shows a single constraint, which states that a person cannot both buy and steal an object, as the activities are mutually exclusive.

Similarly, Table 2 presents beliefs that can arise in this domain. These include observed beliefs (a), such as *(possess Abe cash1 1 2)*, that come from some external source. The table also includes abduced beliefs (b) like *(give Abe Bob cash1 2 3)* and derived beliefs (c) like *(buy Abe*

2. This name indicates its relation to UMBRA (Meadows et al., 2013, 2014), an earlier system that influenced its design, which in turn referred to the shadows on the wall in Plato's Allegory of the Cave.

Table 1. (a) Two definitional rules and (b) one constraint for a simple knowledge base about ways that a person can obtain an object from another person. The definitions specify activities for buying and selling, whereas the constraint states that the two forms of transfer are mutually exclusive.

```
(a) ((buy ?to ?from ?obj ?money ?t2 ?t3) <=
     (possess ?from ?obj ?t1 ?t2) (possess ?to ?money ?t1 ?t2) (money ?money)
     (give ?to ?from ?money ?t2 ?t3) (give ?from ?to ?obj ?t2 ?t3)
     (possess ?to ?obj ?t3 ?t4) (possess ?from ?money ?t3 ?t5)))

    ((rob ?to ?from ?obj ?t2 ?t3) <=
     (possess ?from ?obj ?t1 ?t2) (threaten ?to ?from ?obj ?t2 ?t3)
     (give ?from ?to ?obj ?t2 ?t3) (possess ?to ?obj ?t3 ?t4))

(b) ((obtain ?obj) ::
     (buy ?to ?from ?obj ?money ?t2 ?t3) (rob ?to ?from ?obj ?t2 ?t3))
```

Bob car1 cash1 2 3). We will not go into details about justifications, but these take the form of instantiated definitions, with a derived belief as the head and with observed, derived, or abduced beliefs as antecedents. An explanation is simply a collection of beliefs that are linked through a set of justifications, whereas a world is a set of beliefs connected by zero or more such explanations. As discussed earlier, each relational literal has an associated set of worlds for which that belief does *not* hold. In this case, most beliefs (including all observations) hold for all worlds, giving the empty set, but some are excluded from one alternative or another.

We have not shown the world history for this scenario, but it includes three nodes. One of these, *W*, is the root note from which processing starts. Two other worlds, *W.1* and *W.2*, denote children of the root node that share many beliefs but that differ on a few relations. In particular, the beliefs *(threaten Abe Bob car1 2 3)* and *(rob Abe Bob car1 2 3)* do not hold in *W.1*, whereas *(give Abe Bob cash1 2 3)* and (buy Abe Bob car1 cash1 2 3) are not included in *W.2*. The fact that these two worlds are children of *W* indicates that the root world violated some constraint, which in turn led to its deactivation and to the generation of its successors.

4.2 Processing in PENUMBRA

Like most cognitive systems, PENUMBRA operates in successive cycles, in this case at two distinct levels. On each pass through the top level, the architecture checks the environment for the arrival of newly 'observed' beliefs. Any such elements are added to working memory and associated with all existing worlds. At the second level, PENUMBRA carries out a number of inference cycles that either extend the contents of working memory by adding new beliefs or eliminate inconsistencies among them in newly created worlds. Once the system has executed the specified number of iterations or until nothing has transpired, it returns to the first layer and continues processing. This loop repeats until completing a specified number of observation passes.

The first step of the second-level inference cycle involves selecting a belief on which to focus attention. To make this decision, PENUMBRA invokes heuristics that are determined by the settings for a set of parameters. One of these declares how to filter candidates, say by eliminating ones that, when last serving as a focus, led to no rule application. Another specifies criteria with which to

Table 2. (a) Observed beliefs, (b) abduced beliefs, and (c) derived beliefs from the domain introduced in Table 1, including the worlds in which they do not hold. Each belief comprises a predicate and its arguments, which may be either constants or skolems that start with asterisks.

```
(a) (possess Abe cash1 1 2) [ ]
    (possess Bob car1 1 2)  [ ]
    (possess Abe car1 3 4)  [ ]
    (money cash1)  [ ]
(b) (give Abe Bob cash1 2 3)  [W.2]
    (threaten Abe Bob car1 2 3)  [W.1]
    (give Bob Abe car1 2 3)  [ ]
    (possess Bob cash1 3 *s1)  [W.2]
(c) (buy Abe Bob car1 cash1 2 3)  [W.2]
    (rob Abe Bob car1 2 3)  [W.1]
```

evaluate beliefs, such as their recency or the number of skolems they contain. The final parameter states how to pick a belief from a ranked list, such as taking the best-scoring candidate or using probabilistic sampling based on scores. On each inference cycle, PENUMBRA uses these parameter settings to select a focal belief and continues to the next stage. Nothing forbids selection of the same focus on successive cycles, but typical heuristics encourage attentional change.

The next phase addresses repair, but the elaboration stage is simpler, so we will describe it first. On this step, PENUMBRA finds all definitional rules with antecedents that unify with the focus belief.[3] Each such unification for a rule R provides a set of variable bindings B that let the system find all complete and partial matches of R that are consistent with B. Rules are indexed by predicates that occur in their antecedents, so PENUMBRA ignores any rules that have no connection to the focus. After it has produced a set of matches, the system selects one of them for application, again using parameters to determine details. One such parameter controls filtering of candidates, say by retaining only rule instances in which each element contains a nonskolem argument or that connect at least two existing beliefs. Another indicates how to score remaining matches, say by the number of unsatisfied antecedents or the number of worlds the rule would elaborate. A third parameter specifies how to select a candidate based on these numbers, say by picking the best-scoring alternative or by choosing from a distribution based on the scores.

Once it has selected an instantiated definition, PENUMBRA applies the rule to generate new beliefs. These always include a derived belief based on the head, although this may contain skolem constants if some of the predicate's arguments were unbound in the partial match. In addition, if any of the rule's antecedents were unmatched, the system adds an abduced belief for each omitted element, again possibly with skolem arguments, that serve as plausible assumptions to complete the conditions of the partially matched rule. Moreover, it generates a justification that links the inferred head to the beliefs on which it depends, for possible use in future processing. Also, recall that associated with each belief is a set of worlds in which it does not hold, so PENUMBRA must determine this set for each newly derived or abduced element. To this end, the system retrieves the

3. One might also retrieve rules with matched heads, but the current implementation does not consider this option.

Table 3. PENUMBRA's two-level processing cycle for generating explanations through associative abduction. The pseudocode assumes that definitions and constraints are available as domain knowledge.

```
Explain(N, K)
Let Root be an empty possible world and let Worlds be Root.
Let Beliefs and Justifications each be the empty set.
Let Cycle be 0.
Until Cycle = N,
  [After which return the list of active worlds.]
  Let Beliefs be Union(Observe( ), Beliefs).
  Increment Cycle.
  Let Count be 0.
  Until Count = K,
    Let Focus be Select-Focus(Beliefs).
    Let Conflicts be Detect-Conflicts(Focus, Beliefs, Constraints).
    If Conflicts is not empty,
    Then let Selected-Conflict be Select-Conflict(Conflicts).
         Let Worlds be Resolve(Selected-Conflicts, Justifications, Worlds).
    Else let Matches be Find-Matches(Focus, Beliefs, Definitions).
         Let Selected-Rule be Select-Match(Matches).
         Add Selected-Rule to Justifications.
         Let New be Apply-Rule(Selected-Rule).
         Let Beliefs be Union(New, Beliefs).
    Increment Count.
```

set of worlds attached to each matched antecedent, takes their union, and associates this combined set with each of the new elements. Because beliefs store worlds in which they do not hold, applying definitions can produce new beliefs that appear in narrower contexts but never in broader ones.

Another stage of the cognitive cycle checks for violated constraints. As with definitions, PENUMBRA only considers constraints with at least one element that unifies with the current focus belief. The system finds all such inconsistencies and, if there is more than a single constraint violation, selects one of the conflicts for correction. As before, parameters specify the details of this selection process. One indicates how to score each conflict, say by assigning a uniform value or using the number of competing worlds that a violation discriminates. Another parameter determines how the architecture should use these scores to decide which conflict should receive immediate attention, say at random or by selecting the best-scoring alternative. The system always deals with a detected constraint violation before it uses definitions to elaborate its beliefs, although it must focus attention on a relevant belief to notice an inconsistency.

Having selected an inconsistency between two beliefs, *B1* and *B2*, PENUMBRA resolves it by taking four steps. First, the system deactivates all worlds in which both *B1* and *B2* occur. Next, for each such inconsistent world, *W*, it creates two new child worlds, *W1* and *W2*. PENUMBRA then finds which assumptions belief *B1* depends on uniquely and which ones instead underlie only *B2*, ignoring any that they share. After this, it removes *B2* and its assumptions from *W1*, along with all derived beliefs that follow from them, by adding that world to the sets associated with the beliefs. Similarly, the sytem removes *B1* and its unique assumptions, from *W2*, together with any

Table 4. Eight parameters that PENUMBRA uses to control heuristic search through the space of abductive explanations. Each parameter can take on different settings to produce distinct reasoning behaviors.

Parameter description	*Default setting*
How should focus processing filter beliefs?	has led to some rule application
How should focus processing score beliefs?	recency of belief
How should focus processing select a belief?	prefer higher scores
How should definition processing filter matches?	some nonskolems, two+ matched
How should definition processing score matches?	number of unmatched antecedents
How should definition processing select a match?	prefer lower scores
How should constraint processing score conflicts?	uniform value
How should constraint processing select a conflict?	at random with equal probability

derived beliefs that depend on them. These modified children, which represent two distinct ways to eliminate the detected conflict, become new terminal nodes in the phylogenetic world history.

Not all constraint violations result from problematic assumptions. In some cases, two or more explanations can be derived entirely from observed beliefs and still produce inconsistencies. A classic example involves different parse trees for a given English sentence. In this situation, PENUMBRA still deactivates any inconsistent worlds and creates new children, but it has no abduced beliefs to remove in them. Instead, it simply removes one of the two conflicting beliefs in each child world, along with any derived beliefs that depend on them. The system also adds their negation to the relevant child, with the details of rule selection ensuring it will not be derived again later. Another important situation occurs when an abduced belief conflicts directly with an observed one. In this case, the observation takes precedence, leading to removal of the assumption and anything it supports, rather than to the creation of two active worlds.[4]

PENUMBRA's use of heuristics to guide search, including its reliance on focus beliefs to direct attention, has both advantages and disadvantages. There are obvious efficiency benefits, since the relational pattern matching is an expensive process and the system must only consider candidates with antecedents that unify with the focus. On the other hand, the parameter settings that are responsible for selecing a definition to apply may cause PENUMBRA to overlook some elaborations, so the final worlds obtained may be incomplete. Similarly, a given world may contain violated constraints that remain undetected because the system never attends to any of the beliefs involved, leading to unnoticed inconsistencies. However, humans exhibit kindred limitations and they often manage in complex reasoning situations regardless.

4.3 Implementation and Use Details

We have implemented the PENUMBRA architecture in Steel Bank Common Lisp. The system follows the two-level cognitive cycle for generating abductive explanations summarized in Table 3. This includes the mechanisms just described for selecting focus beliefs, for detecting and eliminat-

4. Approaches that also consider revising observations are certainly possible, as reported by Rose and Langley (1986) in their account of belief revision in the early history of chemistry.

ing constraint violations, and for matching and applying definitional rules. The latter two processes are responsible for creating new worlds and elaborating existing ones, respectively. Both operate over the distributed encoding of worlds reported earlier. PENUMBRA carries out search through a space of plausible explanations, with choices determined by its available knowledge, the observations it encounters, and the heuristics specified in the eight parameters summarized in Table 4. These handle the filtering, scoring, and selection of focus beliefs, the scoring and selection of constraint violations, and the filtering, scoring, and selection of elaborative definitions.

To run PENUMBRA on a given scenario, a user loads a file that contains domain definitions and constraints, along with settings for the system's parameters and a set of predicates that can be assumed during abduction. The file includes an *Observe* function that, on each call, returns a set of observed beliefs that are added to working memory. The user runs the architecture by calling it with two arguments: the number of desired observation cycles and the number of inference steps per observation cycle. The system iteratively invokes the observation function, on each pass repeatedly selecting a focus belief, using definitions to elaborate its current explanations, and repairing constraint violations as it detects them. Upon completion, PENUMBRA returns the list of worlds that remain active and the beliefs associated with each one.

5. Demonstrations of PENUMBRA's Behavior

The theory we presented in Section 3 offers a promising approach to associative abduction, but a primary reason for implementing the ideas in a running sytem is to show that it behaves as intended. Remember that we want to account for the ability to explain observations by linking them through available knowledge, to introduce plausible assumptions about unobserved relations or events, to process observations incrementally upon their arrival, to detect and eliminate conflicting beliefs, and to generate multiple explanations when they are viable. In this section, we report case studies of PENUMBRA's behavior that demonstrate its functionality.

We will focus on two primary domains, plan understanding and sentence parsing, that involve hierarchical structures, support alternative explanations, and involve incremental processing. We demonstrate PENUMBRA's ability to generate single explanations, show it can recover after making incorrect guesses, and handle scenarios in which multiple explanations are consistent with observations. For the runs reported here, we set the system parameters to favor more recently added beliefs unless they had failed to produce a rule application and to calculate, during the elaboration stage, the number of unmatched antecedents for candidate rule instances and prefer those with lower scores, provided the rule had not already been applied earlier. Finally, when choosing a detected constraint violation to repair, we specified that the program should select from among candidates at random with equal probability. These correspond to the default settings listed in Table 4.

5.1 Finding Single Explanations

We designed our initial scenarios to demonstrate that PENUMBRA can generate hierarchical explanations when only a single interpretation suggests itself. One domain revolved around understanding a sequence of observed actions with blocks. Knowledge included two rules for how to construct pyramids, one for lifting a block and stacking it on another and a recursive rule for lifting a block

and stacking it on an existing pyramid. We provided two constraints: a block cannot appear as the top of two distinct pyramids and a block cannot be a pyramid's bottom if it has been stacked. Observations were the action sequence *(pick-up B) (stack B C) (pick-up A) (stack A B)*, where *C* was wider than *B* and *B* was wider than *A*. Taken together, these actions produced a three-level pyramid.

We first ran PENUMBRA on this problem in nonincremental mode, in that we provided all observations at the outset of processing. In this scenario, one observation cycle and three inferences produced 13 beliefs, including a three-level hierarchical explanation of the observed actions. In addition, one constraint check detected a pyramid construction in which the bottom block had also been stacked, leading it to abandon the root world and one of its children. We also presented PENUMBRA with the same observations incrementally, providing one action after another. Here five observation cycles and three inferences found the same hierarchical plan as before, with the program generating 13 beliefs and one consistent world, while abandoning two others for inconsistencies. Along the way, the system predicted the action *(stack A B)*, merging the abduced and observed versions later, which was the main difference from the nonincremental run.

The second domain involved syntactic parsing of declarative English sentences. Knowledge included a context-free grammar with seven definitional rules for inferring parts of speech, along with six higher-level rules for generating adjective phrases, noun phrases, verb phrases, and sentences. Two constraints specified that a given adjective or noun could serve as the head of only one constituent, whereas a third stated that a particular word can have only one part of speech. When given seven observations encoding the sentence *The big dog chased a black cat* in nonincremental mode, PENUMBRA took one observation cycle and 13 inferences to create a five-level parse tree encoded in 20 beliefs, none of them abduced. When presented with these observations incrementally, PENUMBRA required seven observation cycles to produce the same 13 inferences and parse tree. No constraint violations occurred in either run, so all beliefs were associated with the original root world. Both the parsing and pyramid tasks were easy, producing a single interpretation without constraint violations, but they demonstrated the focus and elaboration modules interact as intended.

5.2 Dealing with Garden Paths

We devised additional scenarios to examine PENUMBRA's behavior in misleading situations that require it to detect the problem and recover. In sentence processing, these are known as *garden paths*, but the same challenge can arise in any domain that involves sequential observations. We used the same domains as before, but we provided observations that offered ambiguous interpretations. For instance, in the pyramid domain, we introduced blocks of the same size (*B1* and *B2*, *C1* and *C2*) that could be stacked to produce different constructions. Here we provided the the system incrementally six observations – *(pick-up B1) (stack B1 C1) (pick-up B2) (stack B2 C2) (pick-up A) (stack A B1)* – that interleaved building a three-block pyramid and a two-block one. In this run, PENUMBRA predicted correctly that block *A* would be placed on *B1*, then merged this abduced belief when its analog was observed later. However, the program also guessed incorrectly that block *A* would be placed on *B2*, leading to a conflict between its assumption and observation. This caused it to abandon the world with the abduced belief and retain the one that was consistent with the observed action. This run involved six observation cycles, during which six inferences and two constraint violations produced 24 beliefs and five worlds, four of which PENUMBRA abandoned.

For the parsing domain, we gave the system definitional rules that associated different parts of speech with the same word and thus allowed multiple parses for initial portions of sentences. When given *The old man the boats*, a classic garden path sentence, PENUMBRA infers that the word *old* is an adjective and a noun, but then realizes that it cannot have both parts of speech, leading it to abandon the root world and create two childen, one for each option. Analogous steps happen for *man*, which can be either a noun or verb, but not both at the same time. Along the way, the system identifies *The old man* as a noun phrase, but this never leads to a complete parse, while the other interpretation, which views *The old* as a noun phrase and *man* as a verb, produces a full parse tree with 14 beliefs. PENUMBRA does not explicitly abandon the other explanation, which it retains in a separate world while focusing its reasoning efforts on the alternative. This run involved five observation cycles, during which 13 inferences and two constraint violations produced 22 beliefs and seven worlds, three of which PENUMBRA rejected for inconsistency.

5.3 Generating Multiple Interpretations

Finally, we developed a third set of scenarios to test PENUMBRA's ability to generate alternative explanations for observations that support multiple interpretations. These often occur when key relations or events are unobserved, as in the example from Tables 1 and 2, which suggested two explanations for how the car changed possession. To demonstrate the system's behavior on this scenario, we gave it two definitions from Table 1, one for *buy* and another for *rob*, along with a constraint that an agent cannot both buy and steal an object. A run on the four observations in Table 2 (a) produced two inferences and one constraint violation, giving 14 beliefs and three worlds, including the root, which PENUMBRA discarded. The two active worlds correspond to the alternative explanations discussed earlier for this example.

Multiple explanations do not arise in the pyramid domain as we have defined it, so we devised another task that supports multiple plans for going from an initial location to a destination by traversing intermediate points that are never provided to the system. Here we provided definitional rules for two routes from place *A* to place *D*, one that passes through place *B* and another through place *C*. We also specified a constraint that these alternative routes are mutually exclusive. We presented the system with two observations – the agent's initial and final locations – in addition to information about place names. This run took three observation cycles in which two inferences and one constraint violation generated 14 beliefs and three worlds. PENUMBRA abandoned the root but retained its children, which encoded distinct explanations for how the agent reached *D* from *A*.

In some settings, multiple explanations can arise even when all events are observed, as in sentences with multiple parses. To examine such situations, we augmented the earlier grammar with a definitional rule for the preposition *with*, another for prepositional phrases, and two rules for including them in noun phrases and verb phrases, respectively. We then presented PENUMBRA with the sentence *The cat saw a dog with the binoculars*, for which it found two complete parses, one attaching *with the binoculars* to *dog* and another to *saw*. We also provided a new constraint that told the system these cannot occur together, leading it to create one world for the first and another for the second. The two parse trees share considerable structure, with only a few beliefs differing in the two worlds. This run took eight observation cycles, during which 19 inferences and two constraint violations generated 30 beliefs and five worlds, three of which PENUMBRA deactivated.

5.4 Discussion

Together, these runs offer evidence that PENUMBRA supports the key target abilities related to associative generation of explanations that we identified earlier. The system introduces plausible assumptions as needed to support explanations, incorporates observations incrementally, detects and eliminates inconsistent worlds, and generates alternative accounts that are consistent with observations. The empirical studies also indicate the viability of the basic cognitive cycle, which alternates between selecting a focus belief and selecting a definition or constraint to apply. This approach differs radically from classic production systems, which match all rules' conditions on each cycle before deciding which one to fire. Because it limits retrieval to structures related to the focus belief, PENUMBRA may overlook some inferences and inconistencies, as do humans, but it should scale far better to complex explanations and to large knowledge bases.

As noted in Section 2, the literature on explanatory inference revolves around two main paradigms. In *derivational abduction*, each observation appears as a root node in a proof tree, with different proofs sharing other observations and assumptions as terminal nodes. In *associative abduction*, which is less common, both observations and assumptions appear only as terminal nodes in a proof graph, with derived beliefs serving as nonterminal nodes. Our theory, and its implementation in PENUMBRA, are instances of the second framework, which we hold comes closer to everyday reasoning humans. We will not claim that our approach is superior to derivational accounts, as it seems likely that, for any given domain, we can transform one formulation into the other, giving them the same functionality. Neither would empirical comparisons be appropriate at this stage, as our framework is less mature than the alternative, making it more appropriate to demonstrate basic abilities than to report controlled experiments with quantitative measures.

6. Related Research

Our approach to explanatory inference draws on many ideas from the literature on this topic, with the strongest influence coming from earlier work on associative abduction. The reliance on focus beliefs to direct cognitive attention comes directly from AbRA (Bridewell & Langley, 2011) and its successor UMBRA (Meadows et al., 2014), as does our emphasis on incremental processing. These earlier systems also encoded constraints, but used them only to eliminate inferences that would immediately violate them. Our primary extensions include the ability to entertain multiple worlds, a distributed encoding for these alternatives, recording a world history that traces their evolution, and using constraint violations to drive the repair process, which creates new worlds that revise their parents in nonmonotonic ways. PENUMBRA's predecessors carried out greedy search, with one-step lookahead, through a space of explorations, whereas the new system instead supports a more extensive search that expands a frontier of worlds heuristically.

However, our framework also incorporates insights from many other efforts in both abduction paradigms. There has been general agreement that explanations take the form of proof graphs, which link observations through knowledge, that their construction uses this knowledge to introduce plausible assumptions, and that the problem involves search through a space of candidate explanations. We have also borrowed important ideas from elsewhere in the artificial intelligence and cognitive systems literatures, including claims that:

- *Beliefs are organized into worlds* that share some elements but differ in others to encode alternative models of the environment (Baral, 2003; Fahlman, 2011; Bello, 2012);
- *Cognitive processing occurs in cycles* which retrieve and select among rules that alter the contents of a working memory, as in work on production system architectures (Klahr et al., 1987).
- *Observations are processed incrementally*, which leads to detection of inconsistencies and drives repair of explanations (Eckroth & Josephson, 2014; Molineaux et al., 2012);
- *Explanatory search relies on heuristics* to make the process tractable by selecting among alternatives and allocating cognitive resources (Ng & Mooney, 1990; Rose & Langley, 1986).

Our theory incorporates each of these computational insights, but it combines them in novel ways. The result is a promising account for how cognitive systems can generate plausible explanations of observed situations and events in terms of available knowledge.

One research area that did not influence our efforts directly, but that has many parallels, concerns assumption-based truth maintenance systems (de Kleer, 1986). These combine monotonic inference with nonmonotonic repair, use constraints to detect inconsistencies, organize beliefs into worlds (or *contexts*) in which they hold, and store these worlds in a similar distributed manner. Initial techniques generated all consistent worlds, but later work (de Kleer, 1994) incorporated heuristic search. Although the literature on explanation seldom cites this line of research, it shares many features with associative abduction, and this link deserves more attention in the future.

7. Concluding Remarks

In this paper, we presented a new cognitive systems account of abductive explanation. We distinguished between two formulations of explanatory inference and selected one – associative abduction – in which observations appear with assumptions as terminal nodes in proof graphs rather than as root nodes. We stated our theory in terms of postulates about the representation of knowledge, beliefs, justifications, and explanations, as well as processes that inspect and modify these cognitive structures. We also reported PENUMBRA, an implemented system that embodies these assumptions, and its behavior on a number of explanatory scenarios. Our theory incorporates ideas from earlier research on associative abduction, but extends them in important ways and makes new contributions to the study of cognitive systems. These include the use of constraints to detect inconsistencies, a repair mechanism to eliminate them, a world history that tracks the evolution of competing accounts, and a distributed encoding of worlds that avoids duplication of effort during elaboration and repair.

Nevertheless, there remain important limitations to both the theory and its implementation that we should address in additional research. Our demonstrations have shown that PENUMBRA behaves as intended on reasonable scenarios, but we should test it more thoroughly on more domains and on more complex tasks. Promising testbeds include the Monroe corpus for plan recognition (Blaylock & Allen, 2005) and traces extracted from the classic Heider-Simmel video (Gordon, 2018). We also need to devise and implement heuristics for selecting focus beliefs, choosing definitional rules during elaboration, and deciding which constraint violations to address, then run controlled experiments to determine when they are effective. In addition, we should evaluate PENUMBRA's ability to handle large knowledge bases and complex explanations; its use of focus beliefs should let it scale well to these factors, but this remains a hypothesis that we must test empirically.

Furthermore, we should expand our theory's scope and PENUMBRA's corresponding abilities. The current framework supports plausible reasoning and introduction of assumptions as needed, but it does not rank active worlds. The distributed character of inference means this is not essential for generating explanations, but it is important for decision making, so future versions should include criteria for evaluating explanations, such as coherence (Ng & Mooney, 1990). We should also augment the system to incorporate, and reason over, probabilistic information about knowledge and beliefs, supporting computations similar to those in Markov logic networks (Kate & Mooney, 2009). Finally, a fuller treatment of associative abduction would not only learn these probabilities from experience, but create new hierarchical definitions from successful explanations, which would in turn aid future understanding. These extensions would offer a more complete computational theory of abductive inference and its role in the construction of everyday explanations.

Acknowledgements

This research was supported by Grant N00014-17-1-2434 from the Office of Naval Research, which is not responsible for its contents. We thank Paul Bello, Will Bridewell, Stephanie Sage, and Mohan Sridharan for useful discussions about approaches to abductive explanation.

References

Appelt, D. E., & Pollack, M. E. (1992). Weighted abduction for plan ascription. *User Modeling and User-Adapted Interaction*, *2*, 1–25.

Baral, C. (2003). *Knowledge representation, reasoning and declarative problem solving*. Cambridge, UK: Cambridge University Press.

Bello, P. (2012). Cognitive foundations for a computational theory of mindreading. *Advances in Cognitive Systems*, *1*, 59–72.

Blaylock, B., & Allen, J. (2005). Generating artificial corpora for plan recognition. *Proceedings of the Tenth International Conference on User Modeling* (pp. 179–188). Edinburgh: Springer.

Charniak, E., & Shimony, S. E. (1990). Probabilistic semantics for cost based abduction. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 446–451). Cambridge, MA: AAAI Press.

Darwiche, A. (2009). *Modeling and reasoning with Bayesian networks*. New York, NY: Cambridge University Press.

de Kleer, J. (1986). An assumption-based TMS. *Artificial Intelligence*, *28*, 127–162.

de Kleer, J. (1994). *A hybrid-truth maintenance system* (Technical Report). Xerox Palo Alto Research Center, Palo Alto, CA.

Eckroth, J., & Josephson, J. R. (2014). Anomaly-driven belief revision and noise detection by abductive metareasoning. *Advances in Cognitive Systems*, *3*, 123–142.

Fahlman, S. E. (2011). Using Scone's multiple-context mechanism to emulate human-like reasoning. *Proceedings of the AAAI Fall Symposium on Advances in Cognitive Systems*. Arlington, VA: AAAI Press.

Friedman, S., Forbus, K., & Sherin, B. (2018). Representing, running, and revising mental models: A computional model. *Cognitive Science*, *42*, 1110–1145.

Gentner, D., & Forbus, K. (1991). MAC/FAC: A model of similarity-based retrieval. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 504–509). Chicago, IL: Lawrence Erlbaum.

Gordon, A. (2018). Interpretation of the Heider-Simmel film using incremental Etcetera Abduction. *Advances in Cognitive Systems*, *7*, 23–38.

Hempel, C. G. (1966). *Philosophy of natural science*. Englewood Cliffs, NJ: Prentice-Hall.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., & Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence*, *63*, 69–142.

Kate, R. J., & Mooney, R. J. (2009). Probabilistic abduction using Markov logic networks. *Proceedings of the IJCAI-2009 Workshop on Plan, Activity, and Intent Recognition*. Pasadena, CA.

Klahr, D., Langley, P., & Neches, R. (Eds.) (1987). *Production system models of learning and development*. Cambridge, MA: MIT Press.

Langley, P. (2018). Theories and models in cognitive systems research. *Advances in Cognitive Systems*, *6*, 3–16.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141–160.

Langley, P., Meadows, B., Gabaldon, A., & Heald, R. (2014). Abductive understanding of dialogues about joint activities. *Interaction Studies*, *15*, 426–454.

Makatchev, M., Jordan, P., & VanLehn, K. (2004). Abductive theorem proving for analyzing student explanations and guiding feedback in intelligent tutoring systems. *Journal of Automated Reasoning*, *32*, 187–226.

Meadows, B., Langley, P., & Emery, M. (2013). Incremental abductive reasoning for plan understanding. *Proceedings of the AAAI-13 Workshop on Plan, Activity, and Intent Recognition*. Bellvue, WA.

Meadows, B., Langley, P., & Emery, M. (2014). An abductive approach to understanding social interactions. *Advances in Cognitive Systems*, *3*, 87–106.

Molineaux, M., Kuter, U., & Klenk, M. (2012). DiscoverHistory: Understanding the past in planning and execution. *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems* (pp. 989–996). Valencia, Spain.

Ng, H. T. & Mooney, R. J. (1990). On the role of coherence in abductive explanation. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 337–342). Cambridge, MA: AAAI Press.

Peirce, C. S. (1878). Deduction, induction, and hypothesis. *Popular Science Monthly*, *13*, 470–482.

Raghavan, S., & Mooney, R. J. (2010). Bayesian abductive logic programs. *Proceedings of the AAAI-10 Workshop on Statistical Relational AI* (pp. 82–87). Atlanta, GA.

Reggia, J. A., Nau, D. S. Wang, P. Y. (1985). A formal model of diagnostic inference. I. Problem formulation and decomposition. *Information Sciences*, *37*, 227–256.

Rose, D., & Langley, P. (1986). Chemical discovery as belief revision. *Machine Learning*, *1*, 423–451.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum.

Simon, H. A. (1970). The axiomatization of physical theories. *Philosophy of Science*, *37*, 16–26.

Winston, P. H. (2012). The right way. *Advances in Cognitive Systems*, *1*, 23–36.