Advances in Cognitive Systems X (20XX) 1-6 Submitted X/20XX; published X/20XX

Deductive Reasoning with Incomplete Knowledge via Repeated Analogies

Jason R. Wilson JRW@FANDM.EDU
Lissangel Martinez LMARTIN2@FANDM.EDU
Franklin & Marshall College, 637 College Ave, Lancaster PA 17603

Irina Rabkina IRABKINA@OXY.EDU
Occidental College, 1600 Campus Road, Los Angeles, CA 90042

Abstract

Deductive reasoning is a common human and AI capability. Humans are also capable of reasoning without complete knowledge, but most logic-based systems are limited in their ability to make inferences with incomplete knowledge due to the closed-world assumption. In this paper, we present our approach that uses analogical inferences to make deductive inferences with and without complete knowledge. Our algorithm, Repeated Analogy for Goal Reasoning (RAGeR), like many forward chaining reasoning algorithms makes inferences based on a match between the available facts and the antecedents of the rule. However, RAGeR uses analogical processes to find matching rules instead of unification. We demonstrate that a model of analogical retrieval is effective at identifying applicable models and that RAGeR can make a set of reasonable inferences when given incomplete knowledge. Overall, there is evidence that warrants further exploration in the relation between analogical processes and logical reasoning.

1. Introduction

Humans are remarkably capable of drawing conclusions from a set of premises, even if the set of premises is insufficient to guarantee the conclusion (Johnson-Laird, 2013). However, an incomplete set of premises can prevent conclusions being inferred in many deductive reasoning systems because they rely on the closed world assumption, which stipulates that if a statement is not known to be true then it is assumed to be false. Non-monotonic reasoning helps to resolve this issue by allowing for plausible conclusions to be drawn while recognizing that the conclusions are not infallible (Reiter, 1988). Many approaches to non-monotonic reasoning have been proposed. Circumscription (McCarthy, 1986) and default reasoning (Reiter, 1980) are two approaches based in symbolic logic, but they tend to require information about the set of valid arguments to a predicate or the defaults to assume until contradictory knowledge is made available. More recently, researchers have explored probabilistic models that used conditional probabilities (Pfeifer & Kleiter, 2010) or measures of belief and plausibility (Zhu & Lee, 1993), but these approaches are better equipped to reason about uncertain knowledge than information that is simply unknown.

© 20XX Cognitive Systems Foundation. All rights reserved.

We introduce here a radically different approach to making deductive inferences from incomplete knowledge. In our prior work, we introduced an algorithm that uses repeated analogies to refine a set of observations to predict an agent's goals, where the algorithm may be given an incomplete set of observations (Rabkina et al., 2021). In this work, we propose considerable extensions to the Repeated Analogies for Goal Reasoning (RAGeR) algorithm[1]. The contributions of this work are the following:

- The enhanced RAGeR algorithm that uses a best-first search to explore multiple inference paths.
- Theoretical and experimental evidence that the modified RAGeR performs deductive reasoning, under both full knowledge and incomplete knowledge conditions.
- Evidence that analogical processes may be used to partially emulate unification.

To describe the revised RAGeR algorithm, we proceed as follows. We begin by describing in Section 2 the algorithm and the analogical processes upon which it is built. In Section 3, we test RAGeR using a chemistry domain, experimenting with complete and incomplete knowledge conditions and with a modified process for finding analogous models. We then discuss our results and the limitations of the current implementation in Section 4. Related work is reviewed in Section 5, and we finish by proposing future work in Section 6 and give our conclusions in Section 7.

2. Technical Approach

RAGeR is designed to chain together a series of deductive inferences towards a specified goal. RAGeR does forward chaining by comparing a set of statements to each rule and inferring the consequent when the facts match the antecedents of the rule. Similarly, most logic systems doing deductive reasoning will also infer a consequent of a rule when the antecedents match with some set of statements. Where RAGeR greatly deviates from a typical logic system is how it does the matching. RAGeR uses analogical comparison to find the match, whereas logic systems use unification.

2.1 Background on Analogy

RAGeR uses structure-mapping (Gentner, 1983) as its model of analogy, and relies on MAC/FAC (Forbus et al., 1995) and the Structure Mapping Engine (SME; Forbus et al., 2017) as computational models of analogical processes. These models reason over a set of facts represented in predicate calculus.

MAC/FAC ("many are called, few are chosen") is a model of analogical retrieval. Given a set of facts, MAC/FAC performs a retrieval from a pre-defined set of inference models (deductive rules, each represented as a set of facts) via a two-step process. In the first step, it retrieves the inference models that are most similar to the given set of facts by identifying the models with the most entities (e.g., predicates and constants) in common. This is found by first converting the set of facts and each

1. The acronym of the algorithm has remained the same though the name has been updated to reflect its broader application.

model to a flat vector representations, and then it computes the cosine similarity between the facts vector and each model vector. The three inference models with highest similarity are returned, and passed to the second step.

In the second step, SME is used to find an analogy between each of the three top models and the set of facts. Here, full structured representations are used to construct *mappings* between a set of facts and each model. Mappings are based on the principles of Structure-mapping Theory (Gentner, 1983): *systematicity*, *one-to-one correspondence*, *parallel connectivity*, and *identicality*. Of these, systematicity—a preference for mappings that contain overlaps in higher order structures—and one-to-one correspondence—a hard constraint that states that each entity or variable in the set of facts can correspond to at most one entity or variable in the model—are the most relevant to RAGeR's deductive reasoning abilities.

SME constructs up to three mappings between the given set of facts and each model. Each mapping contains a set of correspondences, a set of analogical inferences, and a similarity score. Correspondences are the expressions and entities that take part in the analogy. Analogical inferences, on the other hand, are projections of parts of the model that do not participate in the analogy. These projections represent expressions that can be added to the set of facts based on the found correspondences. Finally, the similarity score is a measure of how much of the initial set of facts and the model take part in the analogy. Essentially, the higher the similarity score, the better the analogy. We use the similarity score to determine the best mapping from a given retrieval, and as the evaluation function in a best-first search.

Below, we give more detail on how SME and MAC/FAC are used in RAGeR. For implementation details on the SME and MAC/FAC algorithms, please see the original work.

2.2 Single Inference using Analogy

We first describe how RAGeR makes a single deductive inference. At a high level, the process starts by using MAC/FAC to find a model that is analogous to the given set of available facts. Then the analogous model is applied to the set of facts to create a new set of facts. Applying a model to a set of facts derives a new set of facts that includes a newly inferred fact and any unused facts from the parent set. Unused facts are facts in the parent set that were not part of the current inference step. The unused facts are included in the new step so that subsequent inferences may use them.

We define a set of available facts to be a set of logical statements represented in predicate calculus. The set Γ for a simple blocks world scenario may be the following statements:

```
(isOn A B)
(isOn B C)
```

We define a model to be a translation of a Horn clause, where the higher-order relation `hasAntecedent` is used to relate the consequent to each antecedent. For example, the rule

`(isOn X Y)` $\wedge$ `(isOn Y Z)` $\Rightarrow$ `(isAbove X Z)`

would be represented in a model μ with the following statements:

```
(hasAntecedent (isAbove X Z) (isOn X Y))
```

`(hasAntecedent (isAbove X Z) (isOn Y Z))`

Given the set of available facts Γ and the model μ, we want to be able to infer `(isAbove A C)` based on an analogy between Γ and μ. Let us define a model mapping $\mathcal{M}$ to be an analogical mapping between Γ and the model μ. The mapping is found using SME and represents how terms (i.e., predicates, constants, and variables) in the model correspond to terms in Γ. We then *apply* model mapping $\mathcal{M}$ to Γ using the definition in Equation 1 to create a new set of facts based on the consequent of the rule.

$$apply(\Gamma, \mathcal{M}) = \Gamma + \{c|c = consequent(\mathcal{M})\} - \{p|p \in \Gamma \text{ and } p \in antecedent(\mathcal{M})\} \quad (1)$$

The consequent of $\mathcal{M}$ is an analogical inference based on how μ maps to Γ. We define the consequent of $\mathcal{M}$ as the consequent of μ after substituting terms of μ based on the mappings defined in $\mathcal{M}$. In our blocks world example, the mapping is $X \rightarrow A$, $Y \rightarrow B$, and $Z \rightarrow C$. Given this mapping, then the expression `(isAbove X Z)` when mapped to Γ would be `(isAbove A C)`, which is then the inferred consequent. The new set also includes unused facts from the previous set. Unused facts are facts in Γ that are not mapped to an antecedent of the model. Unused facts are then available for subsequent inferences.

2.3 Repeated Analogical Inferences

RAGeR repeatedly applies analogous models to infer new facts. Some fact ϕ is inferred by RAGeR if $\exists\Gamma_i$, $\phi \in \Gamma_i \wedge \Gamma_i \in RAGeR(\Gamma, \mathcal{L})$ where $RAGeR(\Gamma, \mathcal{L})$ is defined in Equation 2.

$$RAGeR(\Gamma, \mathcal{L}) = \Gamma \cup \bigcup_{\Gamma_i \in children(\Gamma, \mathcal{L})} RAGeR(\Gamma_i, \mathcal{L}) \quad (2)$$

This recursive definition states that RAGeR is applied to each child of the initial set of facts Γ. A child is a set of facts derived from a parent set by applying an analogous model to the parent set. Let us define $\mathcal{L}$ as the set of all models. Then the set of all children, formally defined in Equation 3, is the result of finding all analogous models in $\mathcal{L}$ and applying them.

$$children(\Gamma, \mathcal{L}) = \bigcup_{\substack{\alpha \in \\ analogies(\Gamma, \mathcal{L})}} \begin{cases} \{\}, & \text{if } best(\alpha) = \emptyset \\ \{\Gamma'\}, & \text{if } \Gamma - \Gamma' \neq \emptyset, \\ & \text{where } \Gamma' = apply(\Gamma, best(\alpha)) \\ \{\Gamma'\}, & \text{if } \Gamma - \Gamma' \neq \emptyset \text{ and} \\ & second(\alpha) \neq \emptyset, \\ & \text{where } \Gamma' = apply(\Gamma, second(\alpha)) \\ children(\Gamma, \mathcal{L} - model(\alpha)), & otherwise \end{cases} \quad (3)$$

The function $analogies(\Gamma, \mathcal{L})$ returns a set of *analogies*, where each analogy α represents a model μ from $\mathcal{L}$, the set of facts Γ, and how they are analogous to each other. To find which models

in $\mathcal{L}$ are analogous to Γ, we compare Γ with each model in $\mathcal{L}$. The comparison is done using the two phases of MAC/FAC. In the first phase, a quick comparison is made by comparing vectors representing the terms used in Γ and each model. This phase identifies which models are using the most terms in common with Γ. The top three models are then passed to the second phase, which uses SME to do a deeper comparison. For each model, SME attempts to find how the model is analogous to the facts in Γ by finding how terms in the model correspond with terms in Γ while adhering to the principles of structure mapping theory.

To describe how a model μ is analogous to Γ, each analogy α includes one or more model mappings between μ and Γ. Since SME assigns each possible mapping a similarity score, we can rank the possible mappings. The functions $best(\alpha)$ and $second(\alpha)$ return the best and second best mappings available in α, respectively. If a model is analogous to Γ then the matched model has at least one mapping, but it does not necessarily have a second. In this case, the $second$ function would return the empty set.

To demonstrate how RAGeR repeatedly applies analogous models, we extend our blocks world example to show that it can use multiple steps to infer `(isAbove A D)`. Let Γ be the set of facts represented by the following statements:

```
(isOn A B)
(isOn B C)
(isOn C D)
```

We also introduce two additional models. The full set of models in $\mathcal{L}$ is then the following.
μ_1:

```
(hasAntecedent (isAbove X Z) (isOn X Y))
(hasAntecedent (isAbove X Z) (isOn Y Z))
```

μ_2:

```
(hasAntecedent (isAbove X Z) (isOn X Y))
(hasAntecedent (isAbove X Z) (isAbove Y Z))
```

μ_3:

```
(hasAntecedent (isAbove X Z) (isAbove X Y))
(hasAntecedent (isAbove X Z) (isOn Y Z))
```

Given this Γ and $\mathcal{L}$, $RAGeR(\Gamma, \mathcal{L}) = \{\{$`(isOn A B) (isOn B C) (isOn C D)`$\}$, $\{$`(isAbove A C) (isOn C D)`$\}$, $\{$`(isAbove A D)`$\}\}$. The first set, which corresponds to the top node in Figure 1, is the initial set of available facts Γ. The next set (also the middle node in the figure) is the only child of Γ and is the result of applying a mapping from μ_1. The mapping is $X \rightarrow A$, $Y \rightarrow B$, and $Z \rightarrow C$. The consequent is then `(isAbove A C)` and the unused facts in Γ is `(isOn C D)`, which are then the facts in the second set.

The final set is derived by applying a mapping of μ_3 to the second set. The mapping consists of $X \rightarrow A$, $Y \rightarrow C$, and $Z \rightarrow D$. Using this mapping, the consequent is `(isAbove A D)`, which

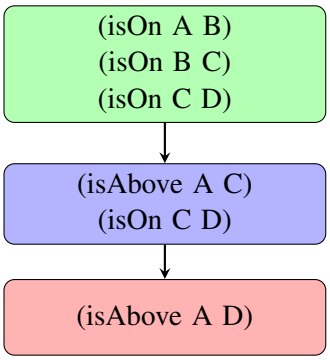

Figure 1. This example shows the inference tree for RAGeR inferring `(isAbove A D)` in two steps.

is the only contents of the final set because all of the facts in the previous set are mapped to some antecedent of μ_3. Since `(isAbove A D)` is in the last set, then RAGeR was able to infer this fact.

We recognize that `(isAbove B D)` is not inferred in this example. There are two equally good mappings of Γ and μ_1. In addition to the mapping described above, there is also the mapping $X \rightarrow B$, $Y \rightarrow C$, and $Z \rightarrow D$, which would lead to the inference of `(isAbove B D)`. The resulting set of facts would then be mapped to μ_2, still resulting in the inference of `(isAbove A D)`. With the two mappings having the same similarity score, RAGeR arbitrarily chooses which it uses. This is a limitation that will be addressed in future work.

2.4 Incomplete Knowledge

Since the analogical mapping between Γ and μ does not require all of the antecedents in μ to be in the mapping, RAGeR is able to apply rules when there is incomplete knowledge. My applying a model where some of the antecedents are not mapped, RAGeR is implicitly assuming those antecedents to be true. If that assumption is incorrect, then the inference could be invalid. On the other hand, not knowing the truth of those antecedents would typically lead to an assumption that they must be false under the closed-world assumption. However, RAGeR does not operate under the closed-world assumption. As a result, it is able to make some inferences with incomplete knowledge.

We consider an example with three blocks, and it is known that `(isOn B C)`. We do not know `(isOn A B)` but would like to infer `(isAbove A C)`. However, RAGeR is not given any knowledge about block A, or even that it exists. As a result, it cannot make the desired inference. However, we can make the desired inference if some information about A is given and that A is shown to have some relation to B and/or C. We continue to withhold the fact `(isOn A B)` while supplementing it with other knowledge. The resulting Γ is the following:

```
(isOn B C)
(onTable C)
(isInStack A)
(isInStack B)
```

```
(isInStack C)
```

Correspondingly, we update model μ_1 to include antecedents regarding the `onTable` and `isInStack` relations. The model μ_1 now is the following:

```
(hasAntecedent (isAbove X Z) (isOn X Y))
(hasAntecedent (isAbove X Z) (isOn Y Z))
(hasAntecedent (isAbove X Z) (onTable Z))
(hasAntecedent (isAbove X Z) (isInStack X))
(hasAntecedent (isAbove X Z) (isInStack Y))
(hasAntecedent (isAbove X Z) (isInStack Z))
```

SME then produces the mapping $X \rightarrow A$, $Y \rightarrow B$, and $Z \rightarrow C$, which RAGeR uses to infer `(isAbove A C)`. This mapping when applied to the first line of μ_1 shows that this inference has the antecedent `(isOn A B)`. Since this statement is not in Γ, this is an implicit assumption that RAGeR makes.

2.5 Best-First Search

A set of facts may have analogies with multiple models. This may especially be the case when there is incomplete knowledge. When multiple models are applied, the set of facts has multiple child sets. Since some of the analogies may be better because there are more similarities between the available facts and the model, we can view the subsequent inferences from those more similar models to possibly be better and worth prioritizing. Adding all children to a priority queue ranked by the similarity of the model used to create the child allows RAGeR to do a best-first search.

When RAGeR is given a goal for which to search, it can conclude the search once that goal is found. A goal is a fact ϕ. Consistent with out definition in Section 2.3, ϕ is inferred if it is in a set in $RAGeR(\Gamma, \mathcal{L})$. If ϕ is not in any of these sets, then RAGeR will run to exhaustion – until no further inferences may be made. In practice, we also introduce a maximum recursion depth to limit computation.

3. Experiments

We demonstrate our approach with a couple of experiments. First, we show the effectiveness of using MAC/FAC to find matching models. Then we compare our algorithm's reasoning chain with and without complete knowledge.

3.1 Problem Domain

We demonstrate our algorithm with problems to infer the classification of chemical elements based on facts about the element's electron configuration and state of matter. To focus our demonstration, we look to infer whether a set of available facts describes an element that either has a positive or negative charge and whether it is a metal, negative metal, nonmetal, negative nonmetal, or noble gas. The models are not complete, as they do no correctly handle some nongaseous negatively charged elements (e.g., carbon), but the models are sufficient to demonstrate chains of reasoning.

In most of the examples below, the initial set of facts Γ is the following set:

```
(isSecondShell (OuterShellFn Silicon) SecondShell)
(shellIsGreaterThanFull (OuterShellFn Silicon) SecondShell)
(isa Silicon Solid)
```

The model library consists of fourteen models. There are three models each for inferring `hasNegativeCharge`, `hasPositiveCharge`, and `hasNeutralCharge`, where each model pertains to a different shell of the atom. There are then five models, one for each goal of whether the initial set of facts represents a `Metal`, `NegativeMetal`, `NonMetal`, `NegativeNonMetal`, or `NobleGas`. Each of the models has a symbol that represents the atomic element that is the subject of the model. The symbol is not formally a variable and RAGeR does not explicitly recognize variables, but the symbol is effectively treated as a variable. Similarly, the symbols `X`, `Y`, and `Z` in the models in Section 2.3 are treated as variables.

3.2 Finding Analogous Models

To test the effectiveness of MAC/FAC, we compare our original algorithm with a variant where SME is run with all models in the library, as opposed to just the top three. As described above we use MAC/FAC to find analogous models in the model library. It operates in two phases, where the first phase does a quick comparison between the facts Γ and each model in the model library $\mathcal{L}$. From this phase, up to three similar models are selected for the next phase. The latter phase, it then uses SME to do a deeper, analogical comparison between the facts and the selected models. We will refer to this approach as "with MAC/FAC". An alternative approach would be to eliminate the first phase and allow SME to do analogical comparisons with all models in $\mathcal{L}$. We will refer to this approach as "without MAC/FAC" since it only uses SME. Comparing with all models would undoubtedly be slower, but one may expect that it may be more accurate since the first phase with its shallow comparison could filter out relevant models. However, our experiment shows that MAC/FAC is faster, has the desired correct inferences, and has fewer inaccurate inferences.

When operating without MAC/FAC, a small change is necessary. Some of the models in $\mathcal{L}$ may have nothing in common with Γ. When then using SME to compare Γ to some model μ, no mappings between Γ and μ could be found. With MAC/FAC, we can assume that the comparisons in the first phase will always yield at least one mapping in the latter phase. Since this assumption is no longer valid, we need to modify the $children(\Gamma, \mathcal{L})$ function to include a case where the mappings are empty. Thus, the condition $\{\}$, if $best(\alpha) = \emptyset$ at the top of Equation 3 was added.

Our experiment starts with Γ, a set of facts that describe silicon. All of the facts necessary to deduce that the element is a negative metal are in the initial set of facts. Thus, this scenario focuses on reasoning with complete knowledge. First, let us examine the inferences made when the algorithm uses MAC/FAC. We see in Figure 2 that after two steps, it infers that the element is a negative metal. The same inference tree is generated regardless of what goal is provided. The inference tree has no branches because the first phase of MAC/FAC always produces exactly one matching model at each step (except the last, where MAC/FAC yields no matching models). As a result, the only inferences made are sound.

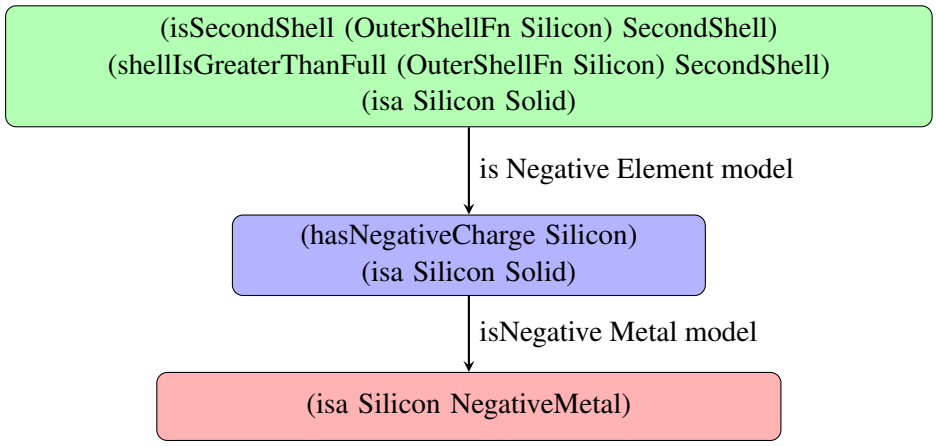

Figure 2. This example shows how RAGeR works when it has complete knowledge. In two inference steps, the only final inference that RAGeR will make is that the element is a negative metal.

We contrast this now with the inferences made without using MAC/FAC. With this variant, any of the five goals can be inferred. In the discussion that follows we abbreviate the goal `(isa Silicon NegativeMetal)` as simply `NegativeMetal`. For `Metal` and `NegativeMetal`, they are inferred directly from the initial set of facts. For `NonMetal` and `NegativeNonMetal`, it first erroneously infers that the element has a negative charge and then makes the goal inference. For `NobleGas`, it first infers that the element has a neutral charge. The inference tree in Figure 3 describes the inferences made towards the goal of `NegativeNonMetal`. The figure also shows inferences relating to the goals `NegativeMetal` or `Metal`, where the subtrees containing Γ and either just the top or top two nodes, respectively, would be generated. The inference trees for `NonMetal` and `NobleGas` are not shown due to their larger size. Notice that, while we demonstrate the fully exhaustive inference trees here, this is not necessarily the case during a reasoning task, when the reasoner might stop after proving a single, possibly erroneous, goal. In other words, the inferences made by the without MAC/FAC variant of our model are not a true superset of those made by the MAC/FAC variant, and may never reach the same conclusions.

All of the inferences in Figure 3 have some problems. The first two inferences that happen in one step require assumptions about the charge of the element to be made. While this is a correct assumption in one case (i.e., `NegativeMetal`), RAGeR makes an invalid assumption about the charge of the element when inferring that it is a `Metal`. For the inference paths leading to `NonMetal`, `NegativeNonMetal`, and `NobleGas`, the incorrect inference about the charge of the element is the result of assumptions that are inconsistent with the initial set of facts. It makes assumptions about the fullness of the element's outer shell while ignoring the fact in Γ stating that the outer shell is more than full. We note that the knowledge that this is an inconsistency (e.g., it cannot be both full and greater than full) is not available to RAGeR, though this is an area of future work that we describe later.

These invalid inferences are the result of bad mappings or unnecessary assumptions that only occur when not using MAC/FAC. SME can produce mappings like $Gas \rightarrow Solid$. This is a valid

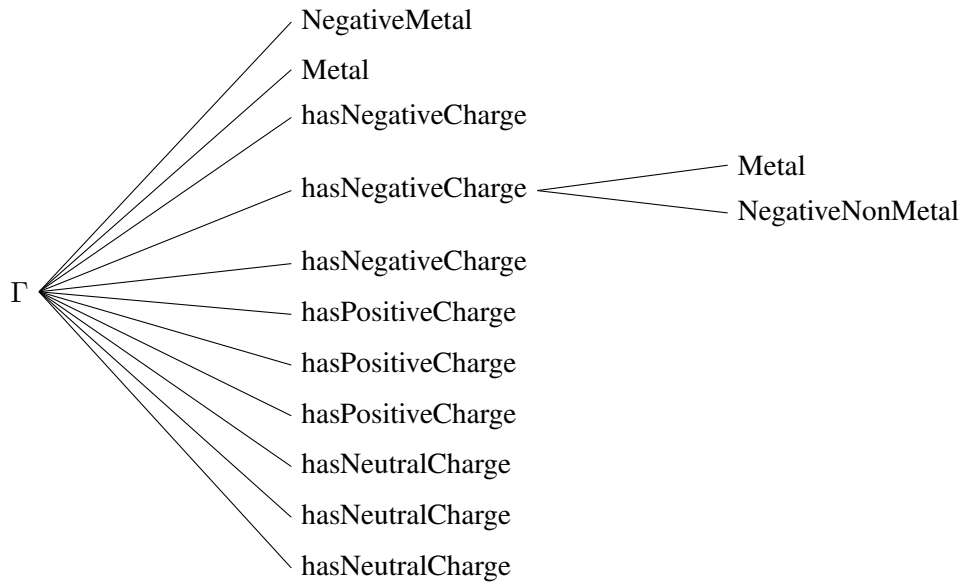

Figure 3. Shows the inference tree for inferring that the initial facts Γ describe a NegativeNonMetal by first inferring that it has a negative charge. The 11 nodes in the middle are made before the two on the right. The fourth node in the middle is the first to be expanded because the model used to make that inference has the highest similarity with the initial facts.

mapping to SME, but it is problematic in our domain. While SME will not make the mapping $shellIsGreaterThanFull \rightarrow shellIsFull$ since it requires the names of predicates to match, it would simply not include the `shellIsGreaterThanFull` predicate in any mapping and then make an assumption with the predicate `shellIsFull`. These mappings and assumptions do not occur when using MAC/FAC because the corresponding models are filtered out as a result of them not having enough in common with the set of available facts. This shows the important role MAC/FAC plays in identifying applicable models.

We also run our two variants (with and without MAC/FAC) to exhaustion, which can be done by providing a goal that is never found. The variant with MAC/FAC still produces the exact same inferences – the inference tree of three nodes shown in Figure 2. However, the inference tree without MAC/FAC is significantly larger. When given a maximum recursion depth of six, the inference tree contains over 900 nodes. At each step, an analogy may be found between the available facts and many of the models, resulting in a possible branching factor of 14, the number of models. Most nodes have fewer than 14 children because no analogy can be made with some set of the models. The lack of analogy with some models is the result of the antecedents of a model having nothing in common with the set of facts. This can be seen in Figure 3 where only eleven models are applied to the initial set of facts.

In the worst case scenario (demonstrated by running each variant to exhaustion), we see that using MAC/FAC is significantly more efficient. The upper bound on the branching with MAC/FAC

is three, meaning that the total inference tree can be no more than 3^n, where n is the number of models. On other other hand, not using MAC/FAC can result in a tree with up to n^n nodes.

3.3 Incomplete Knowledge

When a fact is not available, most logic systems use the closer world assumption to assume that the missing fact is false. However, in real world scenarios missing information could be the result of faulty sensors or information that is unavailable to the reasoning agent. Ideally, unknown facts would not be assumed to be false, which would then allow inferences to be made as if they were true. On the other hand, we also do not want to assume that any expression in the antecedent of a rule could be true, and we instead would only want facts that "fit" with the other available facts to be assumed. We rely on our model of analogy to implicitly find these facts that can be assumed. Generally, assumed facts will have terms in common with the known facts, thereby creating a larger structure of interconnected facts. SME will recognize this larger structure as supporting *systematicity* and thereby allow for RAGeR to make the assumptions.

To show how RAGeR can make inferences without complete knowledge, we compare two scenarios: one in which all necessary facts to infer that the element is a negative metal, and one where the facts that it is solid is missing. The expectation is that in the latter case it will still infer that the element is a negative metal. Since there is missing information, RAGeR may make other inferences based on invalid assumptions. In this case, RAGeR also infers that the element is a negative nonmetal. We show that two different reasoning paths will result in the different inferences based on the implicit assumptions made by RAGeR.

For the scenario in which all facts are known, the inference tree is the same as the previous example (see Figure 2). We contrast this with the inference tree in Figure 4, where the final inferences are that the element is a `NegativeMetal` (for the leaf on the left) or a `NegativeNonMetal` (for the leaf on the right). The implicit assumption made by RAGeR for the `NegativeMetal` inference is that the element is a `Solid` (i.e., the missing fact). The assumption made for the `NegativeNonMetal` inference is that the element is a `Gas`.

4. Discussion

In this work, we present significant updates to the RAGeR (Rabkina et al., 2021) algorithm, which performs deductive reasoning, including from incomplete knowledge. We demonstrate that this approach is robust in a small and limited test domain, where RAGeR must infer the classification of a chemical element given facts about its electron configuration and/or state of matter. RAGeR performs extremely well on this task—especially when using MAC/FAC to retrieve relevant models.

We believe that using MAC/FAC to find the applicable models and identify valid substitutions partially emulates the process of unification. Recall that MAC/FAC is a two-stage retrieval algorithm, with the first stage acting as an efficient initial proxy for structural similarity. It retrieves up to three possible models that are passed to the second stage. The second stage, then, performs full analogical alignment, including proposing analogical inferences. Unification requires matching predicates and consistent variable substitution. Both phases of MAC/FAC contribute to matching predicates. The first phase finds models that includes as many of the desired predicates as possible,

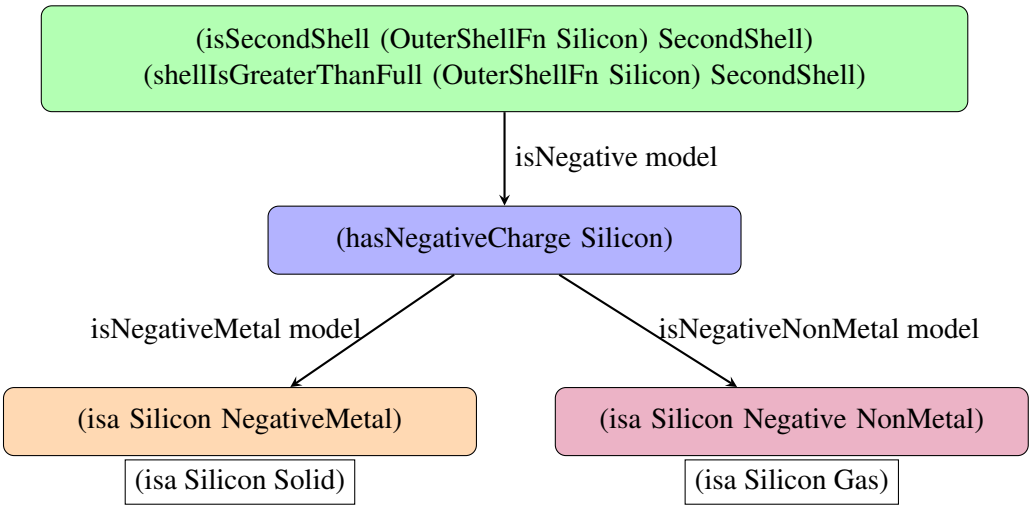

Figure 4. With the initial set of facts missing a crucial piece of information on the state of matter of the element, RAGeR makes two final inferences. Each inference is based on an implicit assumption the RAGeR makes. The assumptions are in the boxes below the nodes at the bottom.

and the second phase uses SME where the identicality principle ensures that identical predicates align. SME also has the one-to-one correspondence principle, which ensures that any entity can be mapped to no more than one other entity. When the base of the mapping is a variable, this represents a consistent variable substitution. Where it differs from unification is that SME does not require all expressions to align and constants in a model may map to a different constant in the set of facts. The first difference is a key reason in why RAGeR is able to reason with incomplete knowledge. The latter difference is a current limitation of the work and will be addressed in future work.

Our experiments show this unification by analogy to be very effective. However, it is not logically sound. That is, when some models are applied, as in Section 3.2, RAGeR can make incorrect inferences. The first phase of MAC/FAC seems to act as a filter in this case, helping limit the models from which inferences are made and thus leading to accurate conclusions. However, it cannot be guaranteed that models leading to unsound inferences will always be filtered out.

The other potential advantage of retrieval via MAC/FAC is efficiency. By limiting the number of models considered by the full analogy stage to three, the number of possible states in RAGeR's best-first search is limited to $O(3^n)$, where n is the number of models. When compared to $O(n^n)$, without the initial filter, this is a substantial improvement in both time and space efficiency. Notably, we do not see an accuracy trade-off. That is, RAGeR with full MAC/FAC retrieval performs at least as well as it does without. In fact, our experiment shows it makes makes fewer faulty inferences. However, this experiment is done on one example, and further work is needed to verify whether the increase in accuracy can consistently be an expected outcome.

While similar patterns of improved efficiency without decreased accuracy due to MAC/FAC retrieval have been found before (Blass & Forbus, 2015), we note that, at least for deductive reasoning, this may not always be the case. While MAC/FAC will, by definition, limit the number of model

explored, it is possible that some appropriate models will be missed (e.g., in domains where multiple models share many similar predicates), causing a decrease in accuracy. This is largely caused by analogy's (and therefore RAGeR's) commitment to structured representations. If predicates are not sufficiently meaningful and/or statements are not sufficiently structured, mistakes can be made. While traditional deductive reasoning systems are also representation-dependent to an extent, so long as predicates are reused between rules and known facts, they are guaranteed to be sound and complete. RAGeR requires much stronger representational commitments, and at present, can make no such guarantees. However, see Future Work, below, for potential improvements.

Another potential limitation of the present work has to do with an implementation choice in RAGeR itself. Specifically, when completing the equivalent of unification, facts that participated in the analogy (i.e., matched to antecedents) are replaced with the inferred consequent. This prevents reasoning contexts from getting too big and the same model from being retrieved repeatedly for the same facts. However, it also limits RAGeR's ability to reuse those facts with other models. For example, given the models:

```
A ∧ B ⇒ C
A ∧ C ⇒ D
```

and the facts

```
A, B
```

RAGeR would replace `A` and `B` with `C`, and potentially be unable to infer `D`. However, this is mitigated by RAGeR's ability to handle incomplete knowledge—removing `A` after it has been used is treated equivalently to not having seen `A` in the first place.

RAGeR's reasoning with incomplete knowledge is not infallible. Its ability to successfully make inferences necessarily depends on the knowledge that *is* available. If too many facts are missing, or remaining facts are too disjoint, no inferences can be made. Furthermore, it is possible for incorrect inferences to be made, given a particularly bad representation or sparse set of facts.

5. Related Work

To the best of our knowledge, there have been two previous implementations that use repeated analogical retrievals for a variant on logical reasoning. Of these, Analogical Chaining (Blass & Forbus, 2017) is the most similar to our approach. Analogical Chaining is a means of abductive reasoning via analogy, used primarily for commonsense reasoning. It makes use of small, causal cases, called Common Sense Units, in its case library to drive reasoning. Common Sense Units are, in principle, similar to the deductive rules used in the current work. However, whereas our rules may contain variables, Common Sense Units are fully grounded. Furthermore, Common Sense Units have a rigid causal structure, whereas ours take the form of Horn clauses.

Like our algorithm, Analogical Chaining builds up from an initial reasoning context by incorporating analogical inferences discovered through repeated analogical retrievals in a best-first search. However, how it incorporates analogical inferences, how it determines which cases are involved in repeated retrievals, and the implementation of best-first search all differ. Specifically:

- **Inference Integration:** Analogical Chaining puts inferences in a separate inference context, and uses a union of the initial reasoning context with the inference context for further retrievals. We, instead, create a new context that replaces the part of the reasoning context used in the match (i.e., the antecedents) with the inference (i.e., the consequent). This makes it less likely that the same case will be retrieved on subsequent retrievals—unless the same rule applies to the reasoning context multiple times.

- **Re-Retrieval:** When no viable analogical inferences are produced from a retrieval, Analogical Chaining re-retrieves with any previously-retrieved cases removed from the case library. We, instead, remove tested mappings, not cases, from consideration. This means that if a retrieved case has multiple good mappings to the reasoning context, they can all be considered.

- **Best-First Search:** Analogical Chaining implements best-first search by backtracking to previous inference contexts when no further inferences can be made (similar to a depth first search, but with ordered children). We, instead, keep a priority queue of generated cases. We believe this to be more robust in open domains, where long chains that do not lead to a viable answer are likely. Both approaches use SME score as the evaluation function.

It is also important to note that Analogical Chaining and RAGeR serve different purposes. Analogical Chaining is intended as an abductive causal reasoner for commonsense. RAGeR, on the other hand, is intended as a general-purpose deductive reasoner. Whether there are empirical differences between them, given the same reasoning task, remains an open question.

Derivational Analogy (Carbonell, 1983) is another approach that uses repeated analogical retrievals to reach a conclusion. However, Derivational Analogy has primarily been used for problem solving (Carbonell, 1985) rather than logical reasoning. It also keeps an explicit history of transformations done to the original case at each retrieval, and stores that history for future use. These histories are then used during subsequent problem solving to support applying the same transformations to a new case. We, on the other hand, do not store explicit histories—although they can be recreated by tracing back through a final case node's ancestors—and instead use analogy for all reasoning. Furthermore, Derivational Analogy has not, to the best of our knowledge, been used for logical reasoning, contrary to the present work.

Other approaches do not use analogy but do allow for open-world reasoning, or reasoning with incomplete knowledge. Some approaches to open-world reasoning use probabilistic frameworks, such as Dempster-Shafer Theory. While it is more commonly used for multimodal fusion given uncertain evidence (Zhao et al., 2022), one approach uses Dempster-Shafer Theory to handle uncertain, unknown, and ambiguous knowledge (Williams & Scheutz, 2016). They define a set of constraints in logical formulae and infer possible bindings to the variables. The likelihood of a variable binding is based on evidence that the binding can satisfy all of the given formulae. In contrast to RAGeR, no new inferences are made. Additionally, like any probabilistic approach, it assumes there is a known probability distribution.To discover these distributions, they typically rely on some amount of training data or assume a uniform distribution. RAGeR however does not require any training data.

There is one approach to making logical deductions from incomplete knowledge that on the surface seems similar to RAGeR. The approach makes logical deductions from incomplete knowledge

by comparing a given set of propositions to a set of rules (Dolzhenkova et al., 2016). If the comparison indicates some commonalities, it determines which antecedents of the rule are not are not known, records them as new truths, and then makes the corresponding inferences. However, there are a number of differences. Most importantly, their approach is only applied to propositional logic, whereas RAGeR is designed for first-order logic. The method of comparison is also drastically different. They use a Disjuncts Division Operation to identify missing literals that are then assumed. Any literal in a rule can be assumed. In RAGeR, on the other hand, assumptions are made only if an analogy between the facts and the rule can be found.

The goals of RAGeR are strongly connected to that of plausible reasoning and defeasible reasoning. Plausible reasoning, for which analogy has often been cited as a component (Pólya, 1990; Collins & Michalski, 1989), makes inferences that may not necessarily be true but it is perhaps more plausible than other inferences. However, it requires a set of *certainty parameters*, one of which describes the degree of similarity. These parameters guide the reasoning in determining which inferences are more plausible. RAGeR instead uses similarity scores produced by SME to determine which inferences are preferred.

Defeasible reasoning is quite similar to plausible reasoning, as it involves making inferences that are compelling, or acceptable, though not necessarily logically sound (Pollock, 1987). Two forms of defeasible reasoning that we briefly introduced above are default logic (Reiter, 1980) and circumscription (McCarthy, 1986). Another type of defeasible reasoning is preferential logic, of which circumscription is a particular case. Preferential logic introduces conditional assertions to describe conclusions that can *normally* be inferred from some premises (Kraus et al., 1990).

6. Future Work

RAGeR can produce unsound inferences when given complete knowledge. The primary reason this can happen is that constants may map to other constants (e.g., $Gas \rightarrow Solid$). Typically, this is a desired feature in analogical reasoning, but constraining this might ensure RAGeR makes sound inferences with complete knowledge. We will add constraints to SME to require that a constant may only map to either a variable or the same symbol in the set of available facts.

There is also a risk that RAGeR produces inconsistent inferences. We saw this when not using MAC/FAC, but this could also happen when using MAC/FAC when given the right set of models. Currently, MAC/FAC does not have a means of recognizing that an element cannot be both positive and neutral or that a block cannot be both above and below another block, for example. RAGeR often produces these inferences when making implicit assumptions. Future work will explore making the assumptions explicit and tracking them with an assumption-based truth maintenance system (ATMS) (de Kleer & Reiter, 1987). Along with the incorporation of *nogoods* (explicit rules about mutually exclusive facts), the ATMS can help identify inconsistent inferences. Additionally, the integration of an ATMS would allow for counterfactual reasoning when operating under incomplete knowledge. In the event that nogoods are not able to be known at the time of reasoning but may later be introduced, RAGeR may need some support for paraconsistent logic, and we will explore how the rules of the active logic machine may be used to control what can be inferred (Anderson et al., 2008).

Lastly, the current heuristic that RAGeR uses to prioritize which set of facts to be the basis of inference next is the SME similarity score that measures the similaritiy between the model used to produce some set of facts and the preceding set of facts. Other heurisitics may be preferred. If quick inference is preferred, then RAGeR should incorporate into the heuristic the number of inference steps made thus far. However, fewer steps may happen as the result of making many assumptions. Thus, an alternative heuristic may be optimizing for fewer assumptions. This heuristic may lead to more inference steps, but fewer assumptions could be an indicator of potentially sound inferences. For the heuristic to take into account an estimated distance to the goal, the heuristic could include symbolic coverage – a count of how many entities in the goal are present in the current case.

7. Conclusion

Repeated Analogy for Goal Reasoning (RAGeR) is an approach to logical deduction that uses analogical processes to identify applicable models and make corresponding inferences. In a small test domain, we show that analogical retrieval using MAC/FAC contributes to making faster and more accurate inferences. Additionally, we show that RAGeR may make inferences when given incomplete knowledge.

References

Anderson, M. L., Gomaa, W., Grant, J., & Perlis, D. (2008). Active logic semantics for a single agent in a static world. *Artificial Intelligence*, *172*, 1045–1063.

Blass, J., & Forbus, K. (2015). Moral decision-making by analogy: Generalizations versus exemplars. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Blass, J. A., & Forbus, K. D. (2017). Analogical chaining with natural language instruction for commonsense reasoning. *Thirty-First AAAI Conference on Artificial Intelligence*.

Carbonell, J. G. (1983). Derivational analogy and its role in problem solving. *AAAI* (pp. 64–69).

Carbonell, J. G. (1985). *Derivational analogy: A theory of reconstructive problem solving and expertise acquisition.*. Technical report, Carnegie-Mellon Univ Pittsburgh PA Dept OF Computer Science.

Collins, A., & Michalski, R. (1989). The logic of plausible reasoning: A core theory. *cognitive science*, *13*, 1–49.

Dolzhenkova, M. L., Meltsov, V., & Strabykin, D. (2016). Method of consequences inference from new facts in case of an incomplete knowledge base. *Indian Journal of Science and Technology*.

Forbus, K. D., Ferguson, R. W., Lovett, A., & Gentner, D. (2017). Extending SME to handle large-scale cognitive modeling. *Cognitive Science*, *41*, 1152–1201.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive science*, *19*, 141–205.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive science*, *7*, 155–170.

Johnson-Laird, P. N. (2013). *Human and machine thinking*. Psychology Press.

de Kleer, J., & Reiter, R. (1987). Foundations for assumption-based truth maintenance systems: Preliminary report. *Proc. American Assoc. for Artificial Intelligence Nat. Conf* (pp. 183–188).

Kraus, S., Lehmann, D., & Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence*, *44*, 167–207.

McCarthy, J. (1986). Applications of circumscription to formalizing common-sense knowledge. *Artificial intelligence*, *28*, 89–116.

Pfeifer, N., & Kleiter, G. D. (2010). Uncertain deductive reasoning. In *The science of reason*, 161–182. Psychology Press.

Pollock, J. L. (1987). Defeasible reasoning. *Cognitive science*, *11*, 481–518.

Pólya, G. (1990). *Mathematics and plausible reasoning: Induction and analogy in mathematics*, volume 1. Princeton University Press.

Rabkina, I., Kantharaju, P., Wilson, J. R., Roberts, M., & Hiatt, L. M. (2021). Evaluation of goal recognition systems on unreliable data and uninspectable agents. *Frontiers in Artificial Intelligence*, *4*.

Reiter, R. (1980). A logic for default reasoning. *Artificial intelligence*, *13*, 81–132.

Reiter, R. (1988). Nonmonotonic reasoning. In *Exploring artificial intelligence*, 439–481. Elsevier.

Williams, T., & Scheutz, M. (2016). A framework for resolving open-world referential expressions in distributed heterogeneous knowledge bases. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhao, K., Li, L., Chen, Z., Sun, R., Yuan, G., & Li, J. (2022). A survey: Optimization and applications of evidence fusion algorithm based on dempster-Shafer theory. *Applied Soft Computing*, (p. 109075).

Zhu, Q., & Lee, E. (1993). Dempster-Shafer approach in propositional logic. *International Journal of Intelligent Systems*, *8*, 341–349.