Exploiting Graph Structure to Summarize and Compress Relational Knowledge

Scott E. Friedman
SIFT, LLC
319 1st Ave North, Suite 400
Minneapolis, MN 55401
friedman@sift.net

Abstract

AI systems consume volumes of relational knowledge to extract patterns and model the world. We face the challenges of scaling to growing volumes of knowledge, identifying more sophisticated patterns and concepts, and compressing and indexing relational knowledge for future access. This paper describes the domain-general *path-specific isomorphic abstraction* (ψ-abstraction) approach for encoding relational knowledge as *relational graphs*, and exploiting paths and cycles in these graphs to guide compression and pattern extraction. ψ-abstraction automatically summarizes relational graphs by (1) identifying cyclic and acyclic paths; (2) generalizing the graph along those paths; (3) explicitly annotating generalizable sequences; (4) encoding qualitative relations over generalized quantity sequences; and (5) optionally pruning generalized knowledge to compress the graph. We demonstrate ψ-abstraction on twelve examples spanning diverse domains: 2D spatial; 3D spatial; temporal (calendar) concepts; abstract feedback cycles; narratives; cell signaling; and anatomy. We evaluate ψ-abstracted knowledge for data compression, pattern identification, and structural similarity of examples within and across categories.

Introduction

Intelligent systems— ranging from higher-order cognitive models, scientific collaboration systems, and domain-specific data miners— consume volumes of relational knowledge to learn informative models of the world and extract patterns of interest. The increasing volume of knowledge presents the first challenge that motivates this paper: addressing scalability demands of relational knowledge to meet storage and efficiency requirements.

A second— and more central— challenge is improving symbolic AI systems' ability to learn sophisticated patterns and categories from examples represented with relational knowledge. For example, symbolic AI systems consume relational knowledge to induce logic programs, perform comparative analysis (e.g., McLure, Friedman, and Forbus, 2015), induce qualitative and quantitative models (e.g., Friedman, Taylor, and Forbus, 2009), and perform case-based reasoning. These techniques compute mappings, hypotheses, and abstractions over *specific* symbols and statements within the relational knowledge, but most of these approaches are less effective at automatically learning and expressing categories involving *arbitrary many* symbols in aggregations, sequences, and cycles. This is a representational (or *re*-representational) challenge for cognitive systems.

This paper presents *path-specific isomorphic abstraction* (ψ-abstraction) designed to address the above scalability and representational challenges. ψ-abstraction traverses relational knowledge as a graph and selectively generalizes isomorphic subgraphs using an enhanced SAGE (Sequential Analogical Generalization of Exemplars) algorithm (McLure, Friedman, and Forbus, 2015). This compactly represents arbitrarily-long relational paths and cycles (e.g., of temporal, causal, flow, or spatial relations) within a matrix of isomorphic entities and values. ψ-abstraction then augments the initial relational knowledge to explicitly describe this matrix as an ordered sequence of entities, and it describes bounded qualitative relationships over generalized qualitative or numerical values. Finally, ψ-abstraction optionally removes the generalized graph structure, leaving the generalized, compressed sequence description and qualitative descriptions.

Rerepresenting the graph with ψ-abstraction jointly improves similarity-based reasoning and compacts the representation. The according claims of this paper are as follows:

1. ψ-abstraction reduces the relational knowledge required to express graphs with repetitive or sequential structure, with potentially lossless compression.
2. ψ-abstraction increases the structural similarity of category exemplars that involve sequences and cycles.

We support these claims with experimental results of ψ-abstraction on fourteen examples spanning diverse domains: 2D spatial representations; 3D spatial representations; temporal calendar concepts; abstract feedback systems; narratives; human anatomy; and cell signalling. We measure information theoretic compression, sequence identification, and the increased structural similarity of within-category examples (e.g., two stacks of blocks, with different height) and across-category examples (e.g., a stack and a ladder) after processing them with ψ-abstraction. Since ψ-abstraction increases within-category and across-category similarity, it is a promising knowledge representation strategy for learning-by-generalization as well as learning with near-misses.

We begin by describing background and related work and then we describe the ψ-abstraction approach and our experiments to support the above claims. We close with a discus-

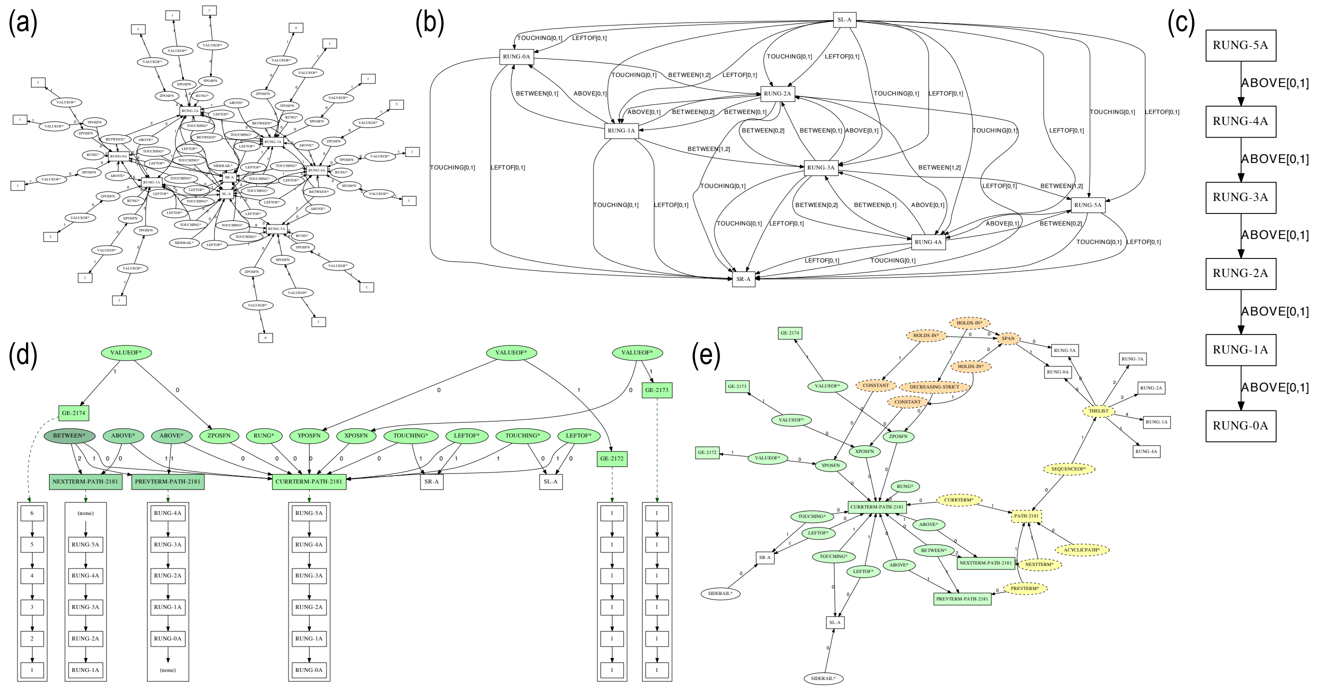

Figure 1: The input, intermediate products, and final product of ψ-abstraction: (a) relational knowledge describing a six-rung ladder displayed as a graph; (b) binary projection of the relational knowledge; (c) binary projection of just the *above* relation; (d) generalization along the *above* binary projection; and (e) ψ-abstracted graph with generalized (green) structure, sequence annotation (yellow), and qualitative descriptions of generalized quantity behaviors (orange).

sion of the implications and future work, including developing additional strategies and constraints for ψ-abstraction.

Background & Related Work

Graph rewriting and graph mining have been research topics in computer science for decades, and these techniques have been used to optimize cybersecurity, compilers, and provenance-tracking.

In the graph compression research, previous work focuses on contracting graphs by pairwise chunking (e.g., Matsuda et al., 2000), which iteratively compresses the graph at single points. By comparison, ψ-abstraction iteratively *generalizes subgraphs* of larger relational graphs along a single path, leaving the vertices behind in case they comprise the subgraph of the next iteration. It later summarizes the generalized subgraphs into a sequential structure.

In graph-mining research, machine learning techniques use kernel methods (e.g., Horváth, Gärtner, and Wrobel, 2004) to allow classifiers to select the most discriminating structures in graph-mining and machine learning settings. Kernel methods and ψ-abstraction both perform subgraph discovery, but ψ-abstraction exploits *paths* of common subgraphs to directly rewrite (and optionally, prune) the graph along paths.

ψ-abstraction builds on previous research on computational models of *structure-mapping* that have produced techniques for describing and quantifying *isomorphic* (i.e., similarly-structured) portions of relational knowledge across cases (Falkenhainer, Forbus, and Gentner, 1989) and within cases (Ferguson, 1994). These techniques are based on the psychological Structure-Mapping Theory of analogy and similarity (Gentner, 1983). Structure-Mapping Theory holds that people match entities and relationships between relational representations (a *base* and a *target* representation) when they reason by analogy, with three constraints:

1. One-to-one: An entity or relation in the base can map to at most one in the target, and the reverse for target-to-base.
2. Parallel connectivity: If a base relation maps to a target relation, so must each of their arguments, in order.
3. Tiered identicality: A relation can only map to another relation with the same predicate.

In graph theory terms, the first (one-to-one) constraint is shared with definition of an isomorphism. The second (parallel connectivity) constraint states that if we treat a relational graph as a directed graph where relations are nodes that have index-labeled edges to their arguments, then if two relation nodes are isomorphic, their child nodes must also be isomorphic. Applied recursively, the first two constraints state that isomorphic (i.e., mapped) nodes must have isomorphic reachable subgraphs.

Computational structure-mapping techniques support analogical generalization (Friedman, Taylor, and Forbus, 2009, Kuehne et al., 2000) and inter-category comparative analysis (McLure, Friedman, and Forbus, 2015) with encouraging results. However, the constraints of structure-mapping, which make it tractable and practical, also present challenges in certain cases. For example, consider a spatial description of a ladder with six rungs and another of a ladder with twelve rungs. Structure-mapping alone may easily map the side-rails of each ladder together, but due to the one-to-one constraint, at most half the rungs on the 12-rung ladder will have corresponding rungs. Intuitively, people probably mentally *rerepresent* the knowledge to account for repetition prior to performing similarity-based reasoning.

Rerepresentation techniques allow us to preserve structure-mapping constraints with more practical results. Previous work in computational rerepresentation has (1) changed the granularity/connectivity of knowledge representation while maintaining semantics (Yan, Forbus, and Gentner, 2003) or (2) relaxed semantics while maintaining granularity/connectivity (Falkenhainer, 1988). However, the implementations of these techniques do not directly solve the ladder example above.

The MAGI algorithm, also based on structure-mapping, detects self-similarity in relational representations by computing and then decomposing global analogical mappings over the entire case/graph (Ferguson, 1994). MAGI may indeed identify the rungs as similar components, but it does not perform the generalization, compression, and qualitative summaries of ψ-abstraction. Further, ψ-abstraction does not need to construct a global mapping, and its time complexity is decoupled from the size of the entire relational description, as we describe below.

Approach

Here we describe the algorithms and representations of ψ-abstraction. While ψ-abstraction is not a cognitive model *per se*, it builds on the same key principle of previous structure-mapping research: reasoning is guided by the [graph-based] structure of relational knowledge. ψ-abstraction employs this principle at a different granularity, by traversing the knowledge graph itself, along paths and cycles, to perform similarity-based generalization and graph summarization.

ψ-abstraction takes structured relational knowledge as input. Consider the predicate calculus statement:

`(valueOf (YPosFn rung-2a) 1).`

This statement includes a relation `valueOf` associating two arguments: a functional term `(YPosFn rung-2a)` referring to the y-position of the entity symbol `rung-2a`; and the number 1. This and other statements can be unambiguously represented in a directed graph. Figure 1(a) illustrates such a graph, describing a six-rung ladder, that shows this relation (at the top of the graph) and others. We refer to this as a *relational graph*.

We draw relational graphs with squares denoting atomic (i.e., symbol or numerical) nodes, ovals denoting non-atomic (i.e., relational or functional) nodes, and top-level assertions marked with * (e.g., the `valueOf` formula is asserted at the top-level but the `YPosFn` formula is not). Edges are labeled by argument index, starting with zero. The relational graph is acyclic, since no formula can contain itself as an argument.

From any given entity, e.g., `rung-2a`, we can identify all statements mentioning it by regressing (i.e., traversing backward) along edges to identify all top-level assertions. From any top-level assertion, we can identify all atomic and non-atomic formulas it includes by traversing forward to compute the reachable subgraph. When we perform these operations in succession, i.e., finding all statements mentioning an entity and then finding all formulas in those statements, we call it the *relevant subgraph* of an entity.[1] Importantly, computing the relevant subgraph of an entity does not involve iterating over all statements, so the time complexity is based solely on the number and structure of statements containing the entity.

Identify the abstraction path. ψ-abstraction first identifies productive paths or cycles to traverse for abstraction purposes. It computes the *binary projection* of the relational graph, as shown in Figure 1(b). This graph is comprised of all entities or functional formulas that are associated with other entities or functional formulas (not including numerical terms) in the relational graph. Edges between the entities are predicate-argument combinations linking the entities. For example, `rung-2a` is linked with `rung-1a` with `above[0,1]` since `(above rung-2a rung-1a)` appears in the relational graph. Similarly, `rung-2a` is linked with `rung-4a` with `between[1,2]` since `(between rung-3a rung-2a rung-4a)` appears in the relational graph.

ψ-abstraction then computes the longest traversal or simple cycle in the graph using any single edge label. In this case, it is the `above[0,1]` label. Figure 1(c) shows the `above[0,1]` subgraph of Figure 1(b). This will be the path ψ-abstraction traverses to absract the relational graph. This is a domain-general heuristic for identifying the abstraction path; however, domain-specific or culture-specific strategies (e.g., following causal links, following temporal relations, or following specific spatial relations) may be used as well, obviating this path-selection step. Note that this is the only time ψ-abstraction accesses or traverses the entire graph; it never accesses the entire graph to perform structure-mapping or subsequent analysis.

Generalize the abstraction path. After ψ-abstraction selects the abstraction path, it iterates over each entry (i.e., vertex) in the path (which may be a cycle). For each entry in the path, ψ-abstraction generalizes the path with the following operations:

1. Identify the next formula f.
2. Compute the relevant subgraph G_f for f.
3. If no generalization exists yet, set the generalization to G_f; otherwise:
   (a) Compare G_f with the existing generalization via structure-mapping to compute the best mapping m, where f is constrained to map to the previous formula.
   (b) Merge G_f into the generalization along m with the SAGE merge operation (McLure, Friedman, and Forbus, 2015).

[1] The result is equivalent to the `CaseFn` of the entity in dynamic case construction (Mostek, Forbus, and Meverden, 2000).

The result of generalizing the relational graph in Figure 1(a) along the path shown in Figure 1(c) is the generalization shown in Figure 1(d). ψ-abstraction generalizations include the generalized relations (shown in green), where brightness indicates frequency. ψ-abstraction generalizations also retain the sequence of values of each generalized entity. For instance, note that the formerly-numerical `valueOf` arguments are generalized entity (i.e., `GE-<nbr>`) nodes. Other generalized entities, e.g., `currterm-path-<nbr>`, have been renamed by ψ-abstraction, which we describe later. Beneath generalized entities are the columns of numerical or symbolic values that assumed that node-space in the graph.

Each row across columns describes a single iteration in ψ-abstraction's generalization, e.g., each rung in the sequence: `rung-5a` has `ZPosFn` 6 and is above `rung-4a`; `rung-4a` has `ZPosFn` 4 and is above `rung-3a`; and so on. We call this an isomorphic matrix, or more succinctly, an *isomatrix*. Each column in the isomatrix is an atomic isomorphism, and each row contains each record/iteration across atomic isomorphisms. This preserves numerical and symbolic data without duplicating the relational data, however it is lossy with respect to retaining which relations are incomplete over which iterations.[2]

Note that some entities, including the siderails `sr-a` and `sl-a` were not generalized against other entities, since they were mapped against themselves for the entire sequence of rungs.

Summarize the abstraction path. ψ-abstraction summarizes the graph based on the generalization and associated isomatrix. It first compares other columns in the isomatrix against the abstraction path. If those sequences can be produced with a single left- or right-shift (for acyclic paths) or a single left- or right-rotation (for cyclic paths), then ψ-abstraction will assert that they are the previous or next term in the sequence. ψ-abstraction renames generalized entity corresponding to the abstraction path `currterm` it renames the generalized entities corresponding to the previous term `prevterm` and next term `nextterm` accordingly, as shown in Figure 1(d).

If there are other columns of generalized entities that are (1) *not* quantity values and (2) *not* next or previous terms, ψ-abstraction returns to its generalization stage and appends this path to its abstraction path, so that each generalization iteration computes the relevant subgraph of the next item in every path sequence. It iterates until no more paths of entities can be generalized.

ψ-abstraction then removes completely-generalized entities, such that no more statements describe the entity in the relational graph that are not included in the generalization, to produce the ψ-abstracted relational graph in Figure 1(e). The ψ-abstracted graph contains generalized (green) structure as well as unabstracted (in white) structure from the original graph. It describes the generalized path(s) with the yellow, dashed summary knowledge in Figure 1(e). This includes annotating which entity is the current (and optionally, next and previous) term, whether the path is cyclic or acyclic, and listing the entities in the sequence.

ψ-abstraction performs ordered qualitative analysis of the various quantity behaviors over the isomatrix, and describes this with the orange, dashed summary knowledge in Figure 1(e). This asserts that for the (sub)sequence `(span rung-5a rung-0a)`, the current term's y-position and x-position are constant, but the z-position is strictly decreasing over the sequence.

This is a more satisfying description of a ladder for exemplar-based learning and reasoning, and ψ-abstraction has constructed it in a domain-general manner, by traversing and generalizing a path in the original relational graph. The isomatrix contains information for reasoning about each rung in isolation, and for partial reconstruction of the original representation if necessary.

Evaluation

We next describe two ψ-abstraction experiments, spanning fourteen examples over multiple domains. We relate each experiment to the overall claims of this paper, including ψ-abstraction's ability to abstract relational knowledge and its effect on structural similarity.

Demonstrating ψ-abstraction across domains. In our first experiment, we ψ-abstract relational graphs in multiple domains to demonstrate ψ-abstraction's ability to summarize different graph structures and multi-sequence expansions. This includes (1) a spatial description of a square spiral, (2) a description of the human circulatory system with qualitative O_2 concentrations from (Friedman and Forbus, 2011), (3) activation of RAF, MEK, and MAPK for cell signaling, and (4) a summarized plot of the children's book "The Spiffiest Giant in Town."

For each input graph, we run ψ-abstraction as described in the previous section. For the circulatory system example and the children's book, ψ-abstraction repeats its generalization stage at least three times, having found more abstractable sequences. Results are shown in Figure 2: Figure 2(top) illustrates the spiral, its relational graph, its generalization and isomatrix, and the resulting ψ-abstracted graph. Figure 2(bottom) illustrates the resulting ψ-abstracted graphs of the remaining three examples. Of the remaining three, only the circulatory system example (Figure 2e) has quantity data, so ψ-abstraction asserts that O_2 decreases from the left atrium (`lt-a2`) to the right ventricle (`rt-v2`), and increases in the complement of the cycle. This is due to O_2 consumption in the body and O_2 infusion in the lungs, respectively.

Three of the ψ-abstracted graphs Figure 2(e-g) contain more than one sequence. This is because ψ-abstraction identified multiple distinct sequences in the isomatrix: in Figure 2(e), ψ-abstraction identified distinct sequences of `FluidFlow` instances, `ContainedFluid` instances, path instances, and container instances; in Figure 2(f), ψ-abstraction identified sequences of activat*ing* entities, activat*ed* entities, `Activation` instances, and base compounds; and in Figure 2(g), ψ-abstraction identifies se-

[2]For instance, if `leftOf` was not asserted for some rung with the right siderail `sr-a`, the isomatrix would not retain *which* rung this was.

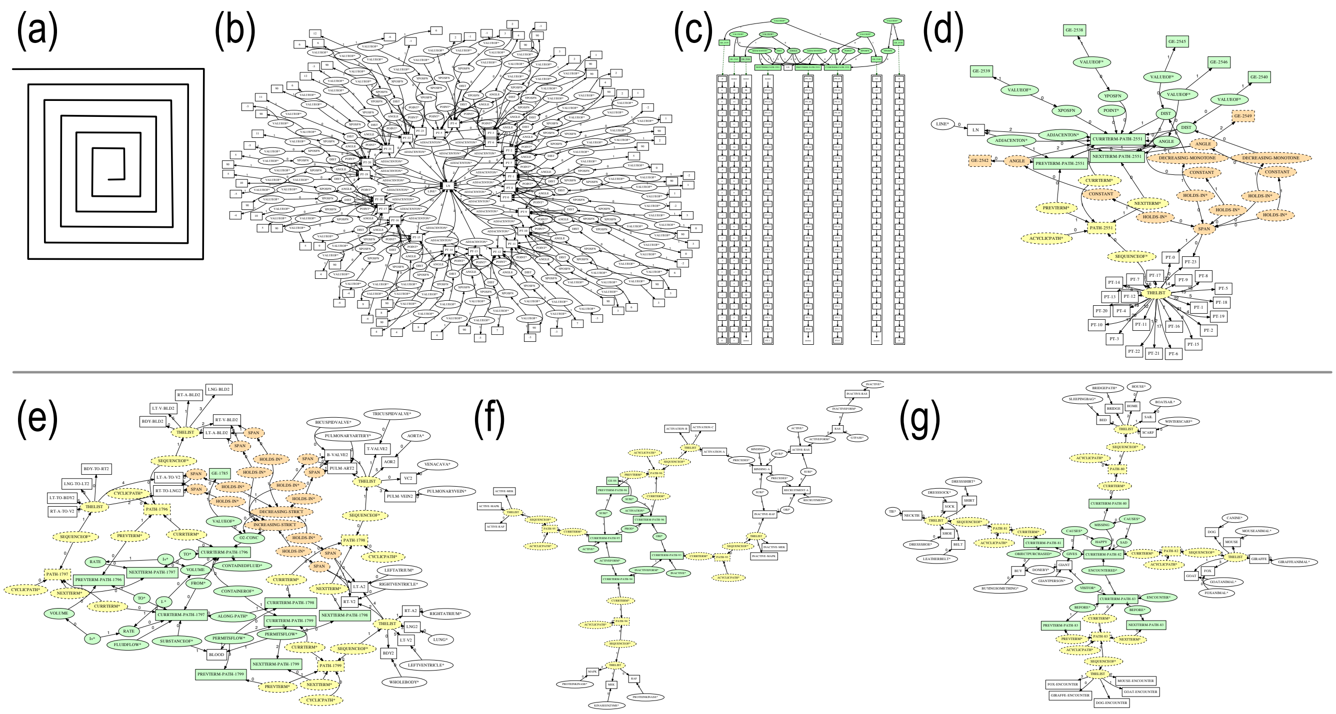

Figure 2: At top, results of abstracting a spatial description of a square spiral: (a) sketch; (b) initial relational graph; (c) generalization matrix; and (d) ψ-abstracted graph. At bottom, three ψ-abstracted graphs: (e) double-loop model of the circulatory system with oxygen concentration; (f) activation of RAF, MEK, and MAPK; and (g) the plot of the children's book "The Spiffiest Giant in Town."

quences of clothing (given by the giant), animals (recipients of the clothing), missing entities (mitigated by the clothing), and the associated `Encounter` events. All of these sequences are joined by generalized structure describing how the different sequences combine semantically.

The ψ-abstracted graphs yielded compression ratios of 5.1 for the spiral, 1.24 for the circulatory system, 1.21 for the cell signaling, and 1.18 for the narrative, including qualitative descriptions and sequence descriptions. Intuitively, the compression rate increases with the length of the sequence, all else being equal.

Evaluating ψ-abstraction's effect on similarity. In the second experiment, we compare pairs of relational graphs with structure-mapping to compute an *original mapping*, and then we compare the respective ψ-abstracted graphs with structure-mapping to compute a *abstracted mapping*. We compare the original and abstracted mappings subjectively for practicality and objectively using normalized similarity scores.

The pairs of examples used for this experiment include: (1) expert vs. novice circulatory system descriptions (without O_2); (2) ladders of different sizes; (3) seasons vs. months of the year with their avg. Minneapolis temperatures; (4) stacks of blocks of different sizes and dimensions; and (5) a four-node feedback cycle of viral videos vs. a three-node feedback cycle of mosquito bites.

Figure 3 illustrates the results in the above order, including the original graphs with an original mapping (at left), and the resulting ψ-abstracted graphs with an abstracted mapping (at right). Many of the original mappings are too large to read (due to having more entities and not as much overlap), but they are included for coarse size comparison. The white nodes of the mapping indicate overlapping/isomorphic structure, and the colored nodes indicate structure from a single (base or target) graph. Boxed colored entities or values indicate that the two entities or values correspond in the mapping at a given `GE-<nbr>` spot.

ψ-abstraction's effect on similarity is most easily noticed in the feedback cycle comparison (Figure 3, bottom): in the original mapping, `(LikesFn video)` and its corresponding qualitative proportionality `qprop` relations map to nothing, due to structure-mapping's one-to-one constraint. In the abstracted mapping, everything corresponds, as a cycle of qualitative proportionality. Similarly, in the seasons-to-months comparison, the abstracted mapping maps the two cyclic sequences, their qualitative temperature changes, and all other nodes except `CalendarSeason` and `CalendarMonth` assertions.

ψ-abstraction increased the similarity of the circulatory mapping by 29.9% (final similarity 0.87), the ladders mapping by 37% (final similarity 1.0), the seasons-to-months mapping by 67% (final similarity 0.98), the stacks mapping by 42.9% (final similarity 1.0), and the feedbacks mapping by 23.4% (final similarity 1.0). We also compared an orig-

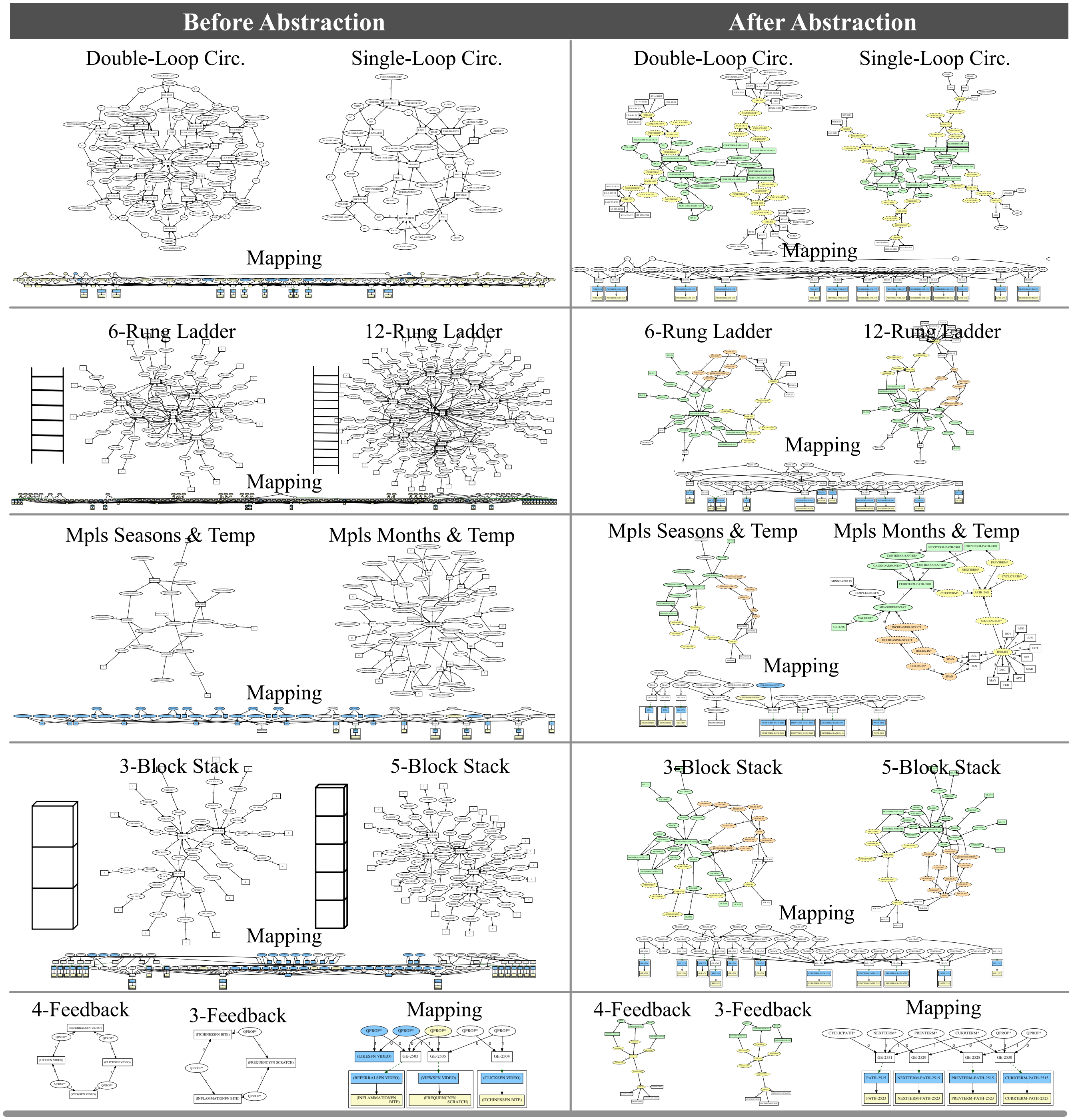

Figure 3: Table of ψ-abstraction results, displaying (at left) the two input representations and their resulting analogical mapping, and (at right) the same inputs after ψ-abstraction, and their resulting analogical mapping. Smaller mappings with more (white) overlapping structure and corresponding (boxed) entities indicate higher similarity.

inal stack and ladder with a ψ-abstracted stack and ladder (i.e., an inter-category comparison), for a similarity increase of 140% (final ψ-abstracted similarity 0.75). This similarity increase is due to structure-mapping utilizing the sequence objects, abstracted category structure, and qualitative descriptions in its mapping rather than putting specific `Block` and `Rung` instances in correspondence.

Conclusion & Future Work

This paper presented the ψ-abstraction approach to generalizing, compressing, and summarizing relational graphs. We have supported our claim that ψ-abstraction compresses sequences of isomorphic subgraphs into summary descriptions on multiple examples across domains. We also supported our claim that ψ-abstraction increases the structural similarity of same-category exemplars when the category permits arbitrary repetition. We reported that ψ-abstraction increases inter-category similarity, e.g., of stacks and ladders. One objection is that ψ-abstraction dilutes the practicality of similarity with its summarization, but recent work in concept learning with *near misses* (e.g., McLure, Friedman, and Forbus, 2015) suggests otherwise, since increasing inter-category similarity makes subtle-but-important differences apparent during comparative analysis. Indeed, ψ-abstraction makes these subtle differences apparent: the siderails of the ladder are unmapped, the fact that the blocks touch (and the rungs don't) is explicit, and the categories (i.e., `Rung` and `Block`) are unmapped. These are logical criteria for differentiating these categories.

In addition to aiding machine learning techniques, ψ-abstraction extracts and describes actionable sequences for goal-directed reasoning. For example, ψ-abstraction describes explicit causal chains for reasoning about intervention, explicit temporal chains for planning and scheduling, and sequential structural relationships, e.g., for determining how to appropriately add more blocks to a stack or continue the pattern of a spiral.

We differentiated ψ-abstraction from other graph rewriting/mining in our discussion of related work, but these might be integrated into ψ-abstraction for breadth and efficiency. For instance, kernel methods that utilize natural graph patterns (e.g., Horváth, Gärtner, and Wrobel, 2004) may efficiently identify or prioritize opportunities for ψ-abstraction.

We demonstrated ψ-abstraction as a proof-of-concept on examples without noisy numerical data or distracting entities and relations, when data in the real world is not so charitable. ψ-abstraction's qualitative summarization is sensitive to some noise by asserting monotonic quantity changes in addition to strict increasing/decreasing changes, but other approaches for computing qualitative behaviors from numerical data (e.g., Žabkar et al., 2013) might be more robust. Aside from the numerical data, distracting entities, e.g., a box the ladder is resting upon, would— at present— interfere with ψ-abstraction's generalization, e.g., by including the box in its path-based compression of the rungs. We believe we could trivially constrain ψ-abstraction's path traversal to only include nodes (i.e., entities) of identical categories. This means ψ-abstraction would traverse each entity along the `above` relation path that shares identical `isa` relations (e.g., `Rung` instances) and stop when it reaches a non-category member. Constraining ψ-abstraction and identifying novel traversal strategies is near-term future work.

References

Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41(1):1 – 63.

Falkenhainer, B. 1988. Learning from physical analogies: a study in analogy and the explanation process. Technical report, DTIC Document.

Ferguson, R. W. 1994. Magi: Analogy-based encoding using regularity and symmetry. In *Proceedings of the 16th annual conference of the cognitive science society*, 283–288. Citeseer.

Friedman, S. E., and Forbus, K. D. 2011. Repairing incorrect knowledge with model formulation and metareasoning. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*.

Friedman, S.; Taylor, J.; and Forbus, K. 2009. Learning naïve physics models by analogical generalization. *Proceedings of the 2nd International Analogy Conference*.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science* 7(2):155–170.

Horváth, T.; Gärtner, T.; and Wrobel, S. 2004. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 158–167. ACM.

Kuehne, S.; Forbus, K.; Gentner, D.; and Quinn, B. 2000. Seql: Category learning as progressive abstraction using structure mapping. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*, 770–775.

Matsuda, T.; Horiuchi, T.; Motoda, H.; and Washio, T. 2000. Extension of graph-based induction for general graph structured data. In *Knowledge Discovery and Data Mining. Current Issues and New Applications*. Springer. 420–431.

McLure, M.; Friedman, S.; and Forbus, K. 2015. Extending analogical generalization with near-misses. *Proceedings of AAAI 2015*.

Mostek, T.; Forbus, K. D.; and Meverden, C. 2000. Dynamic case creation and expansion for analogical reasoning. In *AAAI/IAAI*, 323–329.

Yan, J.; Forbus, K. D.; and Gentner, D. 2003. A theory of rerepresentation in analogical matching. *Proceedings of the 25th Annual Conference of the Cognitive Science Society*.

Žabkar, J.; Možina, M.; Bratko, I.; and Demšar, J. 2013. Learning qualitative models from numerical data. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 3195–3199. AAAI Press.