Supplementary material for :

A Change for the Better? Assessing the Computational Cost of Re-representation

by Todd Wareham, Robert Robere, and Iris van Rooij

In this supplementary material, we provide proofs of the complexity-theoretic statements appearing in our paper. In order to prove intractability results for the input-output mappings associated with the computational-level models of re-representation formulated in our paper, we must formalize the various entities mentioned in these input-output mappings. We first review the graph-based formalization of analogy derivation given in [6] (Section 1). This is followed by a brief summary of techniques for establishing polynomial-time and fixed-parameter intractability (Section 2) and the proofs of the various intractability and tractability results presented in our paper (Section 3).

1 Definitions

Predicate-structures are formalized as a restricted type of vertex- and edge-labeled directed acyclic graph.

Definition 1 *A* **concept-graph** *is a quadruple* $(G, \lambda_A, \lambda_B, \lambda_P)$ *for a directed acyclic graph* $G = (V, A)$ *and functions* λ_A, λ_B*, and* λ_P *called* labelings *such that:*

1. $\lambda_A : A \rightarrow \mathcal{N}$.
2. λ_B *is 1:1 and onto and defined on the leaves of* G.
3. λ_P *is defined on the internal vertices of* G.
4. *If* v *is an internal vertex, then* λ_A *either enumerates the arcs leaving* v *or is constantly* 0 *on this set. In the first case,* v *is* **ordered***, and in the second* **unordered**.
5. *For internal vertices* u, v *with* $\lambda_P(u) \neq \lambda_P(v)$*, the following hold:*
   (a) *Either both* u *and* v *are ordered or* u *and* v *are unordered;*
   (b) u *and* v *both have the same number of children in* G*; and*
   (c) $\{(v', \lambda_A(v, v')) \mid (v, v') \in A\} \neq \{(u', \lambda_A(u, u')) \mid (u, u') \in A\}$.

A concept-graph is ordered if all of its internal vertices are ordered; otherwise, if it contains at least one unordered predicate, it is unordered. Note that in a concept-graph, internal vertices correspond to predicates or functions and leaves correspond to objects, and the labelings λ_P, λ_B, and λ_A assign predicate-types, object-names, and predicate argument-order labels, respectively. For later, let us define P to be the range of λ_P, *i.e.*, P is the set of predicate-types in the concept-graph.

Analogies between pairs of concept-graphs are formalized in terms of a restricted type of subgraph isomorphism. Recall that a graph isomorphism between directed graphs $G = (V, A)$

and $G' = (V', A')$ is a 1:1 onto function $f : V \rightarrow V'$ such that$(u,v) \in A$ if and only if $(f(u), f(v)) \in A'$, and a subgraph isomorphism between directed graphs G and G' is a graph isomorphism f between a subgraph of G and a subgraph of G'. Let $G_f = (V_f, A_f)$ be the directed graph defined by such a subgraph isomorphism f.

Definition 2 *Given a pair of concept-graphs $\mathcal{G} = (G = (V, A), \lambda_A, \lambda_B, \lambda_P)$ and $\mathcal{G}' = (G' = (V', A'), \lambda_{A'}, \lambda_{B'}, \lambda_{P'})$, an* **analogy-morphism** *of $\mathcal{G}$ and $\mathcal{G}'$ is a subgraph isomorphism f which satisfies the following three conditions:*

1. *For all $v \in V_f$, all children of v in G are also in V_f;*
2. *For all internal nodes $v \in V_f$, $\lambda_{P'}(f(v)) = \lambda_P(v)$; and*
3. *For all $(v, w) \in A_f$, $\lambda_{A'}((f(v), f(w))) = \lambda_A((v, w))$.*

Note that the various conditions used by Gentner to define an analogy are encoded in this definition – in particular, the fact that an analogy-morphism is a 1:1 onto function ensures one-to-one correspondence, conditions (1) and (3) ensure parallel connectivity, and condition (2) ensures relational focus. We need to model one analogy including and hence being extension of another analogy as specified in problems GIR[R] and GIR[O]. Given two concept-graphs $\mathcal{G}$ and $\mathcal{G}'$ and two analogy-morphisms f and f' between $\mathcal{G}$ and $\mathcal{G}'$, we say that f' **includes** f if all vertex-mappings between $\mathcal{G}$ and $\mathcal{G}'$ in f are also part of f'. Note that this implies that the subgraphs of $\mathcal{G}$ and $\mathcal{G}'$ mapped by f are in turn subgraphs of the subgraphs of $\mathcal{G}$ and $\mathcal{G}'$ mapped by f'.

Systematicity is assessed using the following function on concept-graphs:

Definition 3 *Given a concept-graph $\mathcal{G} = (G = (V, A), \lambda_A, \lambda_B, \lambda_P)$, a function $pval : P \rightarrow \mathcal{N}$ and positive numbers $trd, lm \in \mathcal{N}$, the* **value** *of $\mathcal{G}$ is $val(\mathcal{G}) = \sum_{v \in V} val(v)$ where $val(v) = match(v) + \sum_{(w,v) \in A} trd \times val(w)$ such that $match(v)$ is $pval(\lambda_P(v))$ if v is an internal vertex and lm if v is a leaf.*

The value of an analogy-morphism f of two concept-graphs is $val(G_f)$, and an optimal analogy-morphism of two concept-graphs is any analogy-morphism f for those concept-graphs that has the maximum value of $val(G_f)$ over all analogy-morphisms for those concept-graphs. Note that the function $val()$ defined above is essentially that used to evaluate systematicity in SME, where $pval()$ is a function assigning match-values to individual predicate-types, trd is the trickle-down factor by which the values of parent-predicates are multiplied when added to compute a child-predicates score, and lm is the match-value of objects (see [2] and [3, p. 158] for details).

Given the above, we can formalize problems ADR,AIR[R], AIR[C], and GDR by substituting concept-graphs for predicate-structures, analogy-morphisms for analogies, and $val(G_A)$ for the systematicity of an analogy A.

2 Proving Intractability

Given some criterion of tractability like polynomial-time or fixed-parameter solvability, we can define the class T of all input-output mapping that are tractable relative to that criterion. For example, T could be the class P of decision problems (see below) solvable in polynomial-time, or FPT, the class of parameterized problems that are fp-tractable. We can show that a particular input-output mapping is not in T (and thus that this mapping is intractable) by showing that this mapping is at least as hard as the hardest mapping in some mapping-class C that properly includes (or is strongly conjectured to properly include) T. For example, C could be NP, the class of decision problems whose candidate solutions can be verified in polynomial time, or a class of parameterized problems in the W-hierarchy $= \{W[1], W[2], \ldots, W[P], \ldots, XP\}$ (see [5] and [1], respectively, for details).

We will focus here on reducibilities between pairs of **decision problems**, *i.e.*, input-output mappings whose outputs are either "Yes" or "No". The two types of reductions used in this paper are as follows.

Definition 4 *Given a pair* Π, Π' *of decision problems,* Π **polynomial-time many-one reduces to** Π *if there is a polynomial-time computable function* f *mapping instances* I *of* Π *to instances* $f(I)$ *of* Π' *such that the answer to* I *is "Yes" if and only if the answer to* $f(I)$ *is "Yes".*

Definition 5 *Given a pair* Π, Π' *of parameterized decision problems with parameters* p *and* p, *respectively, ,* Π **fp-reduces to** Π *if there is a function* f *mapping instances* $I = (x, p)$ *of* Π *to instances* $I' = (x', p')$ *of* Π' *such that (i)* f *is computable in* $g(p)|x|^\alpha$ *time for some function* $g()$ *and constant* α, *(ii)* $p' = h(p)$ *for some function* $h()$, *and (iii) the answer to* I *is "Yes" if and only if the answer to* $I' = f(I)$ *is "Yes".*

A reducibility is appropriate for a tractability class T if whenever Π reduces to Π' and $\Pi' \in T$ then $\Pi \in T$. We say that a problem Π is C**-hard** for a class C if every problem in C reduces to Π. A C-hard problem is essentially as hard as the hardest problem in C.

Reducibilities become particularly useful when the following three properties hold:

1. If Π reduces to Π' and Π is C-hard then Π' is C-hard.
2. If Π is C-hard and $T \subset C$ then $\Pi \not\in T$, *i.e.*, Π is not tractable.
3. If Π is C-hard and $T \subseteq C$ then $\Pi \not\in T$ unless $T = C$, *i.e.*, Π is not tractable unless $T = C$.

These properties are easily provable for many commonly-used reducibilities, including those given in Definitions 4 and 5 above. The first and third of these properties will be used to show intractability below relative to tractable T-classes P and FPT and enclosing but not provably properly enclosing C-classes NP, $W[1]$, and XP. Note that these intractability results hold relative to the conjectures $P \neq NP$ and $FPT \neq W[1]$ which, though not proved, have strong empirical support and are commonly accepted as true within the Computer Science community (see [5], [4], and [1] for details).

In the remainder of these supplementary materials, as we will be using reducibilities which operate on decision problems, we will actually be showing results relative to the decision problems associated with problems AIR, AIR[R], AIR[C], and GDR described in the paper — namely, decision problems which ask if the B', T', and A' requested in these input-output mappings exist. This is acceptable for such suitably-defined decision problems because if such a problem is intractable, then so is the associated input-output mapping (otherwise, any tractable algorithm computing the input-output mapping could be used to construct a tractable algorithm for solving the decision problem, which would contradict the intractability of the decision problem).

3 Proofs of Results

3.1 Intractability Results

All of our intractability results are derived relative to the following decision problems associated with ADR, AIR[R], AIR[C], and GDR as defined in the paper:

ANALOGY DERIVATION WITH RE-REPRESENTATION (ADR)
Input: Unordered concept-graphs B and T, an optimal analogy-morphism A between B and T, a rule-set R, and integers k and c.
Question: Are there concept-graphs B' and T' and an analogy-morphism A' between B' and T' such that (i) B' and T' are derivable from B and T by at most k applications of rules from R and (ii) $val(G_{A'}) - val(G_A) \geq c$.

ANALOGY IMPROVEMENT WITH RULE-GUIDED RE-REPRESENTATION (AIR[R])
Input: Unordered concept-graphs B and T, an optimal analogy-morphism A between B and T, a rule-set R, and integers k and c.
Question: Are there concept-graphs B' and T' and an analogy-morphism A' between B' and T' such that (i) A' includes A, (ii) B' and T' are derivable from B and T by at most k applications of rules from R, and (iii) $val(G_{A'}) - val(G_A) \geq c$.

ANALOGY IMPROVEMENT WITH CONTEXT-GUIDED RE-REPRESENTATION (AIR[C])
Input: Unordered concept-graphs B and T, an optimal analogy-morphism A between B and T, and integers k and c.
Question: Are there concept-graphs B' and T' and an analogy-morphism A' between B' and T' such that (i) A' includes A, (ii) B' and T' are derivable from B and T by at most k context-guided re-representations, and (iii) $val(G_{A'}) - val(G_A) \geq c$.

GENERAL DERIVATION WITH RE-REPRESENTATION (GDR)
Input: Concept-graph T such that $Prop(T) = False$, rule-set R, and integer k.
Question: Is there a concept-graph T' such that (i) T' derivable by at most k applications of rules from R and (ii) $Prop(T') = True$.

All of our intractability results will be derived using by reductions from the following well-known NP-hard decision problem [5, Problem GT19]:

CLIQUE
Input: An undirected graph $G = (V, E)$ and an integer k.
Question: Does G contain a clique of size $\geq k$, *i.e.*, is there a subset $V' \subseteq V$, $|V'| \geq k$, such that for all $v, v' \in V'$, $(v, v') \in E$?

The reductions below will use the following transformations to create concept-graphs from undirected graphs:

- Given an undirected graph $G = (V, E)$, let $C(G)$ be the unordered two-level concept-graph consisting of $|V|$ objects and $|E|$ unordered binary predicates, all of the same type, in which each object corresponds to a unique vertex G and each predicate a corresponds to a unique edge $e = (u, v)$ in E such that the arguments of a are the objects corresponding to the endpoints u and v of e in G.

- Given an undirected graph $G = (V, E)$, let $CE(G)$ be the unordered three-level concept-graph consisting of 2 objects and $|V| + |E|$ unordered binary predicates in which each of the $|V|$ vertex-predicates on the second level corresponds to a unique vertex in G and has as arguments both objects on the first level and each of the E edge-predicates a on the third level corresponds to a unique edge $e = (u, v)$ in E such that the arguments of a are the vertex-predicates on the second level corresponding to the endpoints u and v of e in G. Each edge-predicate has the same predicate type and each vertex-predicate has a type different from all other vertex-predicates and the edge-predicates; let this predicate-type and set of predicate-types be denoted by $PE(G)$ and $PV(G)$, respectively.

Note that $C(G)$ was originally defined in the Supplementary Materials of [6]. In the below, let K_k be the graph consisting of a clique on k vertices.

Lemma 1 CLIQUE *polynomial-time many-one reduces to ADR such that in the constructed instance of ADR, $o = 2$ and the value of k and a are both functions of k in the given instance of* CLIQUE.

Proof: Given an instance $I = \langle G = (V, E), k\rangle$ of CLIQUE, construct an instance $I' = \langle B, T, A, R, k', c\rangle$ of ADR as follows: Let $B = CE(K_k)$ and $T = CE(G)$ such that the vertex-predicate types in B and T are different, *i.e.*, $PV(B) \cap PV(T) = \emptyset$, and all edge-predicates in B and T are of the same type, *i.e.*, $PE(B) = PE(T)$, A be an analogy-morphism between the first-level objects of B and T, $R = \{x \rightarrow y \mid x \in PV(B) \text{ and y} \in \text{PV(T)}\}$, $k' = k$, and $c = val(B) - val(G_A)$. Note that A is an optimal analogy-morphism for B and T and the rules in R can only apply to vertex-predicates in B. This construction can be done in time polynomial in the size of I.

To prove that the construction above is a reduction, we must show that the answer to the given instance of CLIQUE is "Yes" if and only if the answer to the constructed instance of ADR is "Yes". We will do this by proving both directions of this implication separately:

- If the answer to the given instance of CLIQUE is "Yes", there is a subset $V' \subseteq V$ that is a clique of size k in G. Let B' be the version of B in which the types of all k vertex-predicates in B are changed to the types of the k vertex-predicates in T corresponding to the vertices in V', $T' = T$, and A' be the analogy-morphism that maps all of B' to the sub-concept-graph of T' consisting of these re-represented vertex-predicates and their associated edge-predicates and objects. Observe that B' and T' are obtained from B and T by $k = k'$ applications of the re-representation rules in R and that $val(G_{A'}) - val(G_A) = c$, which together imply that the answer to the constructed instance of ADR is "Yes".
- If the answer to the constructed instance of ADR is "Yes", then as R only allows re-representation of predicates in T, $val(G_{A'}) - val(G_A) \geq c$ if and only if A' maps all of B to a re-represented sub-concept-graph of T. Hence, by the construction of B' and T', this re-represented sub-concept-graph of T' corresponds to a subset $V' \subseteq V$, $|V'| = k$ such that all pairs of vertices in V' are connected by edges in G. Such a V' is therefore a clique of size k in G, which implies that the answer to the given instance of CLIQUE is "Yes".

To complete the proof, note that $o = 2$ and $k' = a = k$. ∎

Lemma 2 CLIQUE *polynomial-time many-one reduces to ADR such that in the constructed instance of ADR, $|R| = 1$ and the value of k is a function of k in the given instance of* CLIQUE.

Proof: Given an instance $I = \langle G = (V, E), k\rangle$ of CLIQUE, construct an instance $I' = \langle B, T, A, R, k', c\rangle$ of ADR as follows: Let $B = C(K_k)$ and $T = C(G)$ such that the edge-predicate types in B and T are different, *i.e.*, $P(B) \neq P(T)$, A be an analogy-morphism between the k objects of B and any of the k objects in T, $R = \{P(T) \rightarrow P(B)\}$, $k' = k(k-1)/2$, and $c = val(B) - val(G_A)$. Note that A is an optimal analogy-morphism for B and T. This construction can be done in time polynomial in the size of I.

To prove that the construction above is a reduction, we must show that the answer to the given instance of CLIQUE is "Yes" if and only if the answer to the constructed instance of ADR is "Yes". We will do this by proving both directions of this implication separately:

- If the answer to the given instance of CLIQUE is "Yes", there is a subset $V' \subseteq V$ that is a clique of size k in G. Let $B' = B$, T' be the version of T in which all $k(k-1)/2$ edge-predicates corresponding to edges in G connecting pairs of vertices in V' have been changed in type from $P(T)$ to $P(B)$ and A' be the analogy-morphism that maps all of B' to the sub-concept-graph of T' consisting of these re-represented edge-predicates and their associated objects. Observe that B' and T' are obtained from B and T by $k(k-1)/2 = k'$ applications of the re-representation rule in R and that $val(G_{A'}) - val(G_A) = c$, which together imply that the answer to the constructed instance of ADR is "Yes".
- If the answer to the constructed instance of ADR is "Yes", then as R only allows re-representation of predicates in T, $val(G_{A'}) - val(G_A) \geq c$ if and only if A' maps all of B to a re-represented sub-concept-graph of T. Hence, by the construction of B' and T', this re-represented sub-concept-graph of T' corresponds to a subset $V' \subseteq V$, $|V'| = k$ such that all pairs of vertices in V' are connected by edges in G. Such a V' is therefore a clique of size k in G, which implies that the answer to the given instance of CLIQUE is "Yes".

To complete the proof, note that $|R| = 1$ and $k' = k(k-1)/2$. ∎

Lemma 3 CLIQUE *polynomial-time many-one reduces to AIR[R] such that in the constructed instance of AIR[R], $o = 2$ and the values of k and a are both functions of k in the given instance of* CLIQUE.

Proof: Observe that in the reduction given in Lemma 1, A' includes A. Hence, this reduction is also a reduction from CLIQUE to AIR[R]. ∎

Lemma 4 CLIQUE *polynomial-time many-one reduces to AIR[C] such that in the constructed instance of AIR[C], $o = 2$ and the value of k is a function of k in the given instance of* CLIQUE.

Proof: Observe that in the reduction given in Lemma 1, each vertex-predicate in B and T (including those changed by re-representation rule application) is actually part of a hole-context, in that it is sandwiched between known analogy-matches of the two child first-level objects and potential analogy-matches of the parent third-level edge-predicates. Hence, this reduction is also (modulo modifications to the proof of correctness for Lemma 1) a reduction from CLIQUE to AIR[C]. ∎

Lemma 5 CLIQUE *polynomial-time many-one reduces to GDR such that in the constructed instance of GDR, $o = 2$ and the values of k and a are both functions of k in the given instance of* CLIQUE.

Proof: Given an instance $I = \langle G, k \rangle$ of CLIQUE, construct an instance $I' = \langle T, Pred, R, k' \rangle$ of GDR as follows: Let $T = B$, $R = R$, and $k' = k'$ as defined in the reduction in the proof of Lemma 1 and $Prop$ be the function which checks if a re-represented T completely matches a sub-concept-graph of T as defined in that lemma. The the proof of correctness of this reduction is analogous to that given for Lemma 1. ∎

Lemma 6 CLIQUE *polynomial-time many-one reduces to GDR such that in the constructed instance of GDR, $|R| = 1$ and the value of k is a function of k in the given instance of* CLIQUE.

Proof: Given an instance $I = \langle G, k \rangle$ of CLIQUE, construct an instance $I' = \langle T, Pred, R, k' \rangle$ of GDR as follows: Let T and k' be as in the reduction given in the proof of Lemma 2 and $Pred$ be the predicate which checks if a re-represented T contains B as specified in the proof of that lemma. The remainder of the proof of correctness of this reduction is analogous to that given for Lemma 2. ∎

Result 1 *ADR, AIR[R], AIR[C], and GDR are NP-hard.*

Proof: Follows from the NP-hardness of CLIQUE [5, Problem GT19] and either of the reductions from CLIQUE to ADR given in Lemmas 1 and 2, the reduction from CLIQUE to AIR[R] given in Lemma 3, the reduction from CLIQUE to AIR[C] given in Lemma 4, and either of the reductions from CLIQUE to GDR given in Lemmas 5 and 6, respectively. ∎

Result 2 *ADR and GDR are fp-intractable for parameter-sets $\{o, k, a\}$ and $\{k, |R|\}$.*

Proof: Follows from the $W[1]$-hardness of CLIQUE for parameter-set $\{k\}$ [1] and the reductions from CLIQUE to ADR given in Lemmas 1 and 2 and the reductions from CLIQUE to GDR given in Lemmas 5 and 6, respectively. ∎

Result 3 *AIR[R] is fp-intractable for parameter-set $\{o, k, a\}$.*

Proof: Follows from the $W[1]$-hardness of CLIQUE for parameter-set $\{k\}$ [1] and the reduction from CLIQUE to AIR[R] given in Lemma 3. ∎

Result 4 *AIR[C] are fp-intractable for parameter-set $\{o, k\}$.*

Proof: Follows from the $W[1]$-hardness of CLIQUE for parameter-set $\{k\}$ [1] and the reduction from CLIQUE to AIR[C] given in Lemma 4, respectively. ∎

3.2 Tractability Results

Result 5 *ADR, AIR[R], and AIR[C] are fp-tractable for parameter-set $\{p\}$.*

Proof: The key observation here is that the number of possible concept-graphs that can be generated by re-representation in each of these problems is a function of p. As each predicate in B and T can be re-represented at most once, $k \leq 2p$ and there are at most $\sum_{i=1}^{k} \binom{2p}{i} < \sum_{i=1}^{k} \binom{2p}{k} = k \times \binom{2p}{k} < k(2p)^k$ possible choices of which predicates in B and T to re-represent. In the case of AIR[C], each re-representation choice can be checked to see if it is motivated by matches of predicates one level up (which can be done in time polynomial in the size of B and T), yielding at most $k(2p)^k \leq p(2p)^p$ possible (B', T') re-represented concept-graph pairs.. In the case of ADR and AIR[R], as a re-representation rule can only map between observed predicates, each of these choices has at most $|R|^k \leq (p \times p)^k = (2p)^k$ possible ways in which re-representation rules can be applied to these choices, yielding at most $k(2p)^k \times (2p)^k = k(2p)^{2k} \leq p(2p)^{2p}$ possible (B', T') re-represented concept-graph pairs. As deriving the optimal analogy for a given pair of concept-graphs is fp-tractable for parameter-set $\{p\}$ [7, Result AD5] and the extra conditions on these analogies encoded in problems ADR, AIR[R], and AIR[C] can all be checked in time polynomial in the sizes of B and T, all of these problems are fp-tractable for parameter-set $\{p\}$. ▮

Result 6 *GDR is fp-tractable for parameter-set* $\{p, |R|\}$.

Proof: In the case of GDR, there are at most $\sum_{i=1}^{k} \binom{p}{i} < \sum_{i=1}^{k} \binom{p}{k} = k \times \binom{p}{k} < kp^k$ possible choices of which predicates in T to re-represent. However, as it is conceivable that re-representation rules in this problem may create predicates that are not observed in the given T, the number of ways in which re-representation rules can be applied to these choices can only be simplified to $|R|^p$. Hence, by a slight modification of the algorithm described in Result 5, problem GDR is fp-tractable for parameter-set $\{p, |R|\}$. ▮

Result 7 *ADR and AIR[R] are fp-tractable for parameter-set* $\langle o, |R|, a\rangle$.

Proof: As there are now at most a predicates in B and T at which re-representation can occur and each of these predicate can be re- represented at most once, there are at most $\sum_{i=1}^{k} \binom{a}{i} < \sum_{i=1}^{k} \binom{a}{k} = k \times \binom{a}{k} < ka^k$ possible choices of which predicates in B and T to re-represent. As $k \leq a$ and deriving the optimal analogy for a pair of concept-graphs is fp-tractable for parameter-set $\{o\}$ [6, Result 5], this result follows by a slight modification of the algorithms sketched in the proofs of Results 5 and 6. ▮

Result 8 *AIR[C] is fp-tractable for parameter-set* $\{o, a\}$.

Proof: By the logic in the proofs of Lemmas 5 and 7, there are $< ka^k < aa^a = a^{(a+1)}$ possible re-represented $(B'T')$ pairs, each of whose optimal analogies can be derived in time that is fp-tractable for parameter-set $\{o\}$. Hence, AIR[C] is fp-tractable for parameter-set $\{o, a\}$. ▮

Result 9 *GDR is fp-tractable for parameter-set* $\langle |R|, a\rangle$.

Proof: By the logic in the proofs of Lemmas 5 and 7, there are $< ka^k|R|^k < aa^a|R|^a = a^{(a+1)}|R|^a$ possible re-representations of T, each of which can be checked in polynomial time by $Prop$. Hence, GDR is fp-tractable for parameter-set $\{|R|, a\}$. ▮

References

[1] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer, Berlin, 1999.

[2] K. D. Forbus and D. Gentner. Structural Evaluation of Analogies: What Counts? In *Proceedings of the 11th Annual Conference of the Cognitive Science Society*, pages 341–348. Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.

[3] K. D. Forbus, D. Gentner, and K. Law. MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19:141–205, 1994.

[4] L. Fortnow. The Status of the P Versus NP Problem. *Communications of the ACM*, 52(9):78–86, 2009.

[5] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, San Francisco, CA, 1979.

[6] I. van Rooij, P. Evans, M. Müller, J. Gedge, and T. Wareham. Identifying Sources of Intractability in Cognitive Models: An Illustration using Analogical Structure Mapping. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 915–920. Cognitive Science Society, Austin, TX, 2008.

[7] T. Wareham, P. Evans, and I. van Rooij. What Does (and Doesn't) Make Analogical Problem Solving Easy? A Complexity-Theoretic Investigation. *Journal of Problem Solving*, 3(2):30–71, 2011.