What changes in children's drawing procedures? Relational complexity as a constraint on representational redescription

Steven Phillips, Graeme S. Halford, William H. Wilson

Abstract

Children's ability to modify their drawing procedures changes in their first decade. Young children make size/shape changes and end-of-sequence insertions/deletions of drawing elements. Older children also make middle-of-sequence insertions/deletions and position/orientation changes in drawing elements. Why do modifications occur in this order? We argue that older children's modifications require processing ternary relations, which according to a relational complexity theory, is beyond the working memory capacity of young children.

Introduction: Redescription in children's drawings

Karmiloff-Smith (1986, 1992) hypothesized that, more than just behavioral mastery, cognitive development includes a process of reorganization of learned, internal representations so as to facilitate more flexible, creative behavior. In what she calls the *representational redescription hypothesis*, two (broad) levels of development are postulated. At the *implicit* level, knowledge in the system is usable in a very narrow context. At the *explicit* level, such knowledge is reformatted so as to be accessible by other cognitive processes for use in other domains.

One source of support for representational redescription was found in a study on children's drawing procedures. Karmiloff-Smith (1990) studied two groups of children on their ability to modify their drawing procedures. The younger (4-6 years) and older (8-9 years) groups were asked to draw three types of objects: a man, a house, and an animal. Both groups were then asked to draw a man, a house and an animal that does not exist (e.g., a man with two heads). The second task probes the sorts of changes children can make to their "normal" drawing procedures in producing "nonexistent" objects.

Karmiloff-Smith observed four types of changes: (1) *Size/shape change* (4-6 years): Younger children modified their drawings by changing the size or shape of one of its elements (e.g., a man with a square head). (2) *End-of-sequence deletion/insertion* (4-6 years): Elements were deleted/inserted elements at the end of a drawing after which no more elements were drawn (e.g., a house with no windows, where windows are the last elements of a normal drawing procedure). (3) *Middle-of-sequence deletion/insertion* (8-9 years): Older children inserted elements between other elements in their drawing procedure (e.g., man with two heads). (4) *Position/orientation change* (8-9 years): Elements were placed in different positions or orientations (e.g., rotated head).

Although a notion of redescription is intuitively appealing, providing a mechanism has proven difficult. Progress has been hampered by a lack of details about the nature of the process. In particular, if redescription is to be a theory of cognitive development it must explain why the observed changes to drawing procedures occur in the order they do (Boden, 1994). For example, why do children have the capacity to make end-of-sequence insertions and deletions before they have the capacity to make middle-of-sequence insertions and deletions?

Karmiloff-Smith (1990) explains the difference in terms of serial constraints and interrupts. The execution of young children's procedures is constrained to operate in a fixed, serial order. Middle-of-sequence insertions/deletions require interrupting that procedure to insert a new subprocedure. But this explanation almost begs the question by appealing to terms so closely related to descriptions of the data.

Cognitive development is replete with observations of task orderings. We have explained a number of these observations in terms of relational complexity: the maximum number of interacting dimensions of information that must be processed in a single decision (Halford, 1993; Halford, Wilson, & Phillips, in press). Briefly, the capacity to process higher arity relations increases with age: unary relations (at median age one year); binary (two years); ternary (five years); and quaternary (11 years). Therefore, tasks requiring higher arity relations appear later.

Relations are ubiquitous in psychological models, and have a formal definition in computer science (see Appendix A). In this paper, we argue that relational complexity also explains the order of drawing modifications: middle-of-sequence deletions/insertions and position/orientation changes performed by older children require ternary relational information, which exceeds the binary relational information limit of young children's working memory capacity (Halford, 1993; Halford et al., in press).

Relations, associations and explicit/implicit dimensions of variation

Informally, a relation is an abstraction of the world that identifies "connections" between its entities and the roles the entities play within these connections.

So, for example, a relational model of concepts *John loves Mary* and *Sue loves Tom* identifies entity *John* as connected to *Mary*, and *Sue* as connected to *Tom*; and *John* and *Sue* as playing a common (*agent*) role, and *Mary* and *Tom* as playing a common, but distinct (*patient*) role. A system implementing a relational model must respect entity-entity and entity-role connections in some manner.

Relations are conventionally depicted as tables. Connected entities are placed in common rows, and columns identify their roles. Thus, it is natural to talk of relations as structured representations with one or more dimensions (columns) of variation. Since relations may identify connections between entities of the same type (e.g., *taller-than*), each dimension must be uniquely identifiable. Since relations permit retrieval of entities via their roles, each dimension must be *explicitly* identified by the representing system (i.e., accessible to processes within the system). Thus, we say that relations involve representing and processing one or more *explicit* dimensions of variation.

Associations, by contrast, are less structured. An associative model consists of cue-target pairs that permit retrieval of the target given the cue. For example, a square is the cue to press a button. The only requirement for a system implementing an associative model is that distinct entities are uniquely identifiable. There is, in fact, no requirement that the entity's role (position) within the pair be identifiable, since retrieval is unidirectional: a cue can be used to retrieve a target, but a target cannot be used to retrieve a cue, unless the reverse association has already been stored (learned). Associations may also be *configural*, where the combination of entities is the cue to another entity (e.g., red square is the cue to press a button, but blue square is the cue to pull a lever). (Associations may also be chained to generate a sequence of actions by having the target of one pair be the cue to another pair.)

Although configural associations, like relations identify connections between multiple entities, there are two important differences[1] that bear on our analysis of representational redescription. Firstly, configural associations are still unidirectional: cues conjointly permit access to the target, but the target does not permit access to any of the cues. Although face recognition involves relationships between facial features, the process of recognition is essentially a configural association from features to name. It is, in general, not possible to give a name and recall a particular feature. Of course, reverse associations can be represented with configural associations, but implementing omnidirectionality with associative systems requires additional training. With relational systems no additional training is required. Omnidirectionality is a kind of generalization that comes with relational, but not associative systems.

[1]See, also, Phillips, Halford, and Wilson (1995) for behavioural implications.

Secondly, although cues may vary along several dimensions in configural associations, the dimensions themselves are not relevant to the process of retrieval. In our shape-colour example (above), it is not by the fact that blue lies on the colour dimension that elicits the "pull lever" response, but by the fact that it *is* blue. Although a stimulus may be decomposable along many feature dimensions, the values of these dimensions do not play a role in the process (i.e., they are epiphenomenal). Rather, it is the values of the cues themselves that are critical[2]. Thus, we say that (configural) associations involve processing over one or more *implicit* dimensions of variation.

In short, we make a distinction between explicit and implicit dimensions of variation that is crucial to our arguments on cognitive load and redescription. Relational processes operate over explicit dimensions, whereas for associative processes the dimensions are implicit. This distinction is not arbitrary, but has behavioural implications. Relational systems permit omnidirectional access to representational components without additional training. In this sense, omnidirectionality is a type of generalization.

Relational complexity, cognitive load and transitive inference

Relational complexity is the number of explicit dimensions of variable information in a task that must be processed in order to make a decision. Intuitively, a decision based on one variable requires less "mental effort" than one based on two variables, which requires less effort than one based on three variables. Consider, for example, the mental effort required to purchase land. If price is the only relevant variable then the decision is simple. The decision becomes increasingly more difficult when one must integrate land prices with other factors such as area and accessibility.

We say that there is a limit to the number of explicit variables that can be considered at any one time. If that limit is exceeded then performance is degraded to the point of increased errors, or failures to complete the task. A task is a series of necessary decisions (i.e., each decision is a precondition to the next decision). Thus, the complexity of the task is the decision with the highest complexity. In some cases, the complexity of a task can be reduced by using an alternative strategy that relies on making lower complexity decisions. Chunking is one possibility (Halford et al., in press). In the "land purchase" example, price and area variables can be reduced to a single price/area variable on which to make the decision. However, chunking may result in loss of information necessary for making a decision (e.g., knowing

[2]Equally, it is not by virtue that a shark has colour grey and length 2 m that elicits fear, but by the fact that it is a shark.

land costs \$10,000/m^2 does not tell us the total cost or size).

With regard to cognitive development, the observed clustering of cognitive abilities centers on common relational complexity in each task. Studies have shown that many tasks are not performed until some median age. We say that tasks in common age groups share the same relational complexity. So, other tasks with the same complexity should impose the same difficulty for the same group of children.

One *benchmark* task is transitive inference: For example, if John is taller than Mary, and Mary is taller than Mark, then John is also taller than Mark (i.e., $aRb \wedge bRc \rightarrow aRc$). Transitive inference is known to cause great difficulty for children below the median age of five years (see Halford, 1993, for a review). The difficulty of transitive inference has been attributed to the *premise integration* step, when two instances of a binary relation aRb and bRc are combined into an ordered triple $aRbRc$ from which the first and third elements are recovered to make the inference aRc (Halford, 1993; Andrews, 1997; Halford et al., in press). From this and many other tasks, children below the median age of five are said to be limited to processing at most binary relations. Therefore, tasks requiring lower complexity (binary) relations are attained *before* tasks requiring ternary relations (e.g., transitive inference). It is worth recounting two points pertaining to transitive inference as they will be relevant to our arguments regarding the difficulty of various drawing modifications.

Premise integration: ternary

Typically, in transitive inference tasks, subjects are given a number of premise pairs from which they must determine whether a test pair preserves the ordering specified in the premises. In an experimental situation, order information may be encoded as relative height or length of premise materials. For example, subjects are given several two-block premise towers, and asked to determine whether a two-block test tower preserves the height ordering specified in the premise towers. Individual blocks, identifiable by colour, correspond to premise elements (A, B, C, etc), and the position of each block within a tower corresponds to its position in the premise (e.g., red-above-green and green-above-blue towers encode premises (A,B) and (B,C), respectively).

In general, computing a transitive inference requires integrating two binary relational instances (premises) into an ordered triple. Since the assignment of block colour to structural elements (e.g., A, B, etc) is arbitrary, subjects cannot rely on a single premise to determine the correct ordering of the test pair. The premise pair (red, blue) does not guarantee the correct order of a test pair as (red, green), since the occurrence of premises (green, brown) and (brown, red) would imply the order (green, red). Subjects must form an ordered triple to determine the correct ordering of the test pair. And, since every colour (except "end" colours - see below) can be assigned to each of the three positions in the triple, we say the complexity of this task is ternary (3 dimensions of variation). (Joining the two premises is not considered forming an ordered 4-tuple because the two middle positions are always the same, and therefore convey no additional information.)

Labeling (end-point) strategy: binary

A special case of this task can be solved by only considering a single premise pair. In any finite ordering, the two elements at each end of the order will appear in only one of the premise pairs. End points can be identified by counting, which requires no more than a binary relation (i.e., Next(X,X+1)). (Although counting can be thought of in terms of addition, which has three dimensions of variation [i.e., X+Y=Z], one of these dimensions is constant [i.e., X+1=Y] so only has two dimensions of variation.) Suppose premise pairs (red, blue), (green, yellow), (yellow, white), and (blue, green); and test pair (white, green). White occurs only once in the premise list, and is identified as an end element. All that remains is to check whether the test pair appears as one of the premises. Since this operation requires retrieving only one premise at a time, the complexity of this special case is binary.

Complexity of children's drawing modifications

The purpose of this section is to show how some drawing modifications are analogous to particular forms of transitive inference, therefore imposing the same degree of difficulty in terms of relational complexity. For completeness, relational complexity is also identified for simpler drawings. Analysis is presented in order of lowest to highest complexity. The main point is that changes fall into two groups: those that require processing ternary relations, being difficult for young children; and those requiring binary relations or less (i.e., unary or nullary), attainable by young children.

Young children's drawing procedures are characterized as implicit; proceeding in one fixed in direction; and its components are otherwise inaccessible to other cognitive processes (Karmiloff-Smith, 1990). These characterizations are captured by a system of chained associations. An associative system represents the world by pairs of system states A $\rightarrow$ B, such that given (cue) state A, the system generates (response) state B. States A and B may include sensory states elicited by external input; motor commands resulting in drawing fragments; or internal system states. Finite-state automata (discrete states) and simple recurrent networks (continuous states) (Elman, 1990) are formal examples of such systems.

When the appropriate state pairs are represented, associative systems capture the implicit level of representation. For example, a person-drawing procedure may be represented as: Man0 $\rightarrow$ head $\rightarrow$ body $\rightarrow$ arms $\rightarrow$ legs $\rightarrow$ end, where each element (e.g., head) performs two functions: it generates the appropriate motor command, and is a cue for the next motor command. This association-based representation is implicit: no drawing element is directly and independently accessible; and has a fixed order: each element can only be executed by executing its predecessor. (Although a child's drawing procedure would involve many more primitive motor commands, this simplified model suffices to illustrate the characteristic features of an implicit, sequential representation.)

With respect to the issue of relational complexity each element is a constant; there are no dimensions of variation. Since there are no dimensions of variation, we say the complexity of the normal drawing task by executing this procedure is nullary (0-arity), and therefore within the capacity of young children. What makes this system associative and not binary relational is the flexibility of access. Binary relations permit predecessor states to be accessed by their successors (see Phillips, Halford, & Wilson, 1995, for a comparison of associative and relational systems).

Size/shape change based alterations: Unary

Size and shape changes to drawing elements require some way of accessing alternative drawing subprocedures for components (e.g., large head) without having to execute a particular preceding subprocedure. A natural way of (re)representing a drawing procedure is to group subprocedures belonging to common drawing components: Man1 $\rightarrow$ Head $\rightarrow$ Body $\rightarrow$ Arms $\rightarrow$ Legs $\rightarrow$ end, where, for example, the element Head $= \{(\text{head}), (\text{head}'), (\text{head}'')\}$ is a set of possible head-drawing subprocedures. Although there is the same fixed order of execution[3], an alternate subprocedure can be selected from the relation it belongs to. Since each set has only one dimension of variation (whose values are specific subprocedures), the complexity of this modification is unary.

End-of-sequence deletions and insertions: Binary

Children's drawing procedures have a structure analogous to transitive inference tasks: elements in both tasks have an ordering[4]. Both tasks can be represented by a binary relation between previous and next elements. As such, the modifications of deletion and insertion are analogous to certain types of transitive inferences.

Suppose the ordered sequence $\langle$A,B,C,D,E$\rangle$, represented by the binary relation $R(\text{Pred}, \text{Succ}) = \{(\text{A,B}), (\text{B,C}), (\text{C,D}), (\text{D,E})\}$. Deleting element C to form a new subsequence $\langle$B,D$\rangle$ is equivalent to constructing the ordered pair (B,D) by joining pairs (B,C) and (C,D) into the triple (B,C,D), and selecting the first and third elements of the triple. In this case, deleting an intermediate element requires ternary relational information.

In the case where the deleted element(s) comes at the end of the sequence, such modifications are analogous to transitive inferences where the test pair contains an end element. If we assume that a drawing procedure contains an "end-of-sequence" subprocedure ($\dashv$) for terminating the drawing, then deleting end elements only requires constructing a subsequence containing the current and terminating elements (e.g., $\langle$C,$\dashv\rangle$). We noted previously that in the special case of transitive inference where the constructed (test) pair contains an end element the complexity of the task is binary. Since termination of a procedure is the same process regardless of the drawing (i.e., stop drawing), there is no need to identify its value by joining intermediate relations. Thus, end-of-sequence deletion is also binary.

End-of-sequence insertions are not modifications of drawing procedures, as such, since the insertions are performed only after completing a "normal" drawing. Thus, the drawing procedure itself remains intact. Since adding a new item does not require respecting any particular ordering, the complexity of this operation is at most binary (i.e., one dimension for selecting a drawn element plus one dimension for selecting a new element).

Middle-of-sequence deletions and insertions: Ternary

As explained above, middle-of-sequence deletions are analogous to making transitive inferences where the inferred pair does not contain an end element. Thus, the complexity is ternary.

Middle-of-sequence insertions require placing a new element between two ordered elements. This operation cannot be performed by constructing a single ordered pair. Suppose a subject intends to insert a new element F between adjacent elements C and D to construct the new subsequence $\langle$C,F,D$\rangle$. The adjacent elements are represented by the binary relational instance (C,D). A new ordered pair can be constructed combining the first element of the pair (C) with the new element (F) to form the pair (C,F). However, one cannot construct the second ordered pair (F,D) without considering what element precedes F, since the order (F,D) is not among the original pairs. The operation requires integrating the original ordered pairs (C,D) and the inserted pair (C,F) to form the ordered triple (C,F,D). This operation is analogous to a transitive inference except that the resulting join occurs at the first element of the triple, rather than the middle element. Again, the complexity is ternary.

[3] This form of representation does not permit deletion or insertion of elements.

[4] Elements of a drawing procedure are ordered in time, while transitive inference tasks usually have a spatial ordering (e.g., height or length).

Position and orientation changes: Ternary

The older group of children also made position and orientation changes to elements of a drawing, whereas the younger children did not. These modifications can be explained in terms of a series of middle-of-sequence deletions and insertions. For example, modifying the sequence $\langle$A,B,C,D,E$\rangle$ to $\langle$A,B,D,C,E$\rangle$ is achieved by first deleting C from the subsequence $\langle$B,C,D$\rangle$ to generate $\langle$B,D$\rangle$ and then inserting C into the subsequence $\langle$D,E$\rangle$ to generate $\langle$D,C,E$\rangle$. Similarly, orientation changes can be regarded as reordering a subsequence of elements. For example, instead of a head being drawn as the sequence $\langle$face, eyes, nose, mouth$\rangle$, an upside-down head may be drawn as the sequence $\langle$face, mouth, nose, eyes$\rangle$, which also requires a combination of middle-of-sequence deletions and insertions. Thus, both sorts of modifications have a complexity that is ternary.

Discussion: Implications

If relational complexity provides a good account of the development of children's drawings then a number of implications arise. Firstly, transitive inference is attained at the median age of five years, whereas Karmiloff-Smith's data show that only a small percentage ($<$ 10%) of 4-6 years olds make the same sorts of modifications as the 8-10 years. This difference may be explainable in terms of methodology. Early work on transitive inference identified the transitionary period at around seven to eight years of age (Piaget, 1957), which is in line with the drawing data. Yet, subsequent methodological refinements reduced the age to five years (see Halford, 1993). Similarly, refinements to the drawing paradigm may reduce the age of specific modifications. However, what should remain the same for the relational theory to hold is the ordering of modifications (i.e., one should not find evidence of ternary tasks before binary).

Secondly, if young children have the capacity for binary relations one would expect the ability to construct subsequences of arbitrary order by the outer join (juxtaposition) of two unary relations. For example, suppose the unary relation Body-part(P) = {head, body, arm, leg}. Taking the outer join of this unary relation with itself generates all possible pairs of elements. Reiterated, it results in all possible lists of elements. Yet, no evidence of such ability was demonstrated in young children.

However, the experimental paradigm of free drawing doesn't really test this possibility. Although, Karmiloff-Smith did explicitly ask young children to draw *a man with two heads* (at which they failed), their failure can be attributed to the overriding constraint of having to draw a *man*, than the constraint of having to draw *two heads*. Rather, a better test of their capacity to access subprocedures is to ask subjects to draw, for example, *two heads*, or *a head and a leg*. This task frees the subject from the constraint of a particular ordering. Drawing single components requires only unary relations, and conjunctions only binary relations[5]. Both of which should be within the capacity of young children. But, modifications that preserve some part of the original order (e.g., insertions, where the first and third elements in the new subsequence were the first and second elements of the original subsequence) may involve ternary relations.

General discussion: Mechanisms for change

Our primary concern in this paper was to explain the order children's drawing modifications so as to constrain possible mechanisms. Unfortunately, the notion of redescription itself does little to characterize the mechanism. Connectionists routinely regard internal activations and weights as forms of representation. And so, it is only a short step to regard any type of learning rule that changes those internal values as a instance of redescription. But, clearly, there is more intended in redescription than just learning (Clark & Karmiloff-Smith, 1993). We said that Karmiloff-Smith's implicit/explicit distinction corresponds to a difference between associative and relational representational systems, and that the order of change is constrained by the relational complexity of the modification task. Although we have not provided mechanisms for change, our formalization allows us to be a little more specific about the process. Our approach is to identify connectionist networks with the properties formalized in associative and relational systems, and then recast the problem as that of identifying mechanisms for going from one network to the other.

Feedforward (and recurrent) networks are natural candidates for associative systems within the connectionist framework. The association (a, b) is represented by weighted connections between a group of input units, whose activations constitute the vector $\vec{a}$ (representing entity a); and a group of output units, whose activations constitute the vector $\vec{b}$ (representing entity b). The weight matrix W represents the association (a, b) by taking the outer product of cue and target vectors (i.e., $W = a \otimes b$). A target is retrieved by applying a cue at the inputs, and multiplying it with the weight matrix. For example, assuming othonormal vector representations (i.e., perpendicular and unit length), $a \odot W = a \odot a \otimes b + a \odot c \otimes d = b + 0 = b$. Feedforward networks are in general unidirectional, since $b \odot W = b \odot a \otimes b + b \odot c \otimes d = 0$. Again, one can represent the reverse association, but this requires an additional training step (i.e., $W = W + b \otimes a$). In the case of configural associations, where there may not be a linear mapping from cue to target, a group of nonlinear hidden units is added, which is sufficient to represent most functions

[5]Conjunctions of n elements do not imply n-ary relations. Apparently, high arity relations may be reducible to lower arity relations, contingent on an analysis of functional dependencies between relational elements (see Phillips, 1997; Halford, Wilson, & Phillips, in press, and Appendix A, for examples).

(Hornik, Stinchcombe, & White, 1989). Analysis of hidden unit activations has revealed considerable internal structure, with dimensions within the activation space coding for particular features (e.g., Elman, 1990). But, it is the analysis technique that makes explicit a dimension of variation. Thus, feedforward and recurrent networks capture the properties of associative systems, and the implicit level of representation.

Relations are a type of structured object. A general connectionist method for representing structured objects is the tensor network (Smolensky, 1990). In a tensor network, units are connected so as to implement the inner and outer product operators for tensors of arbitrary rank. Assuming the appropriate units and connectivity, there are two ways of representing relations with tensor networks. Suppose the *loves* relation, with instances: *John loves Mary* and *Sue loves Tom*. One method is to take the sum of the outer products of vectors representing entities[6] and their roles. So the *loves* relation is represented by the tensor $T_{JLM} = J \otimes A + L \otimes R + M \otimes P$, and the tensor $T_{SLT} = S \otimes A + L \otimes R + T \otimes P$, where A, R and P are vectors representing the *agent*, *relation* and *patient* roles, respectively. A second method (Halford, Wilson, Guo, Gayler, Wiles, & Stewart, 1994) is to take the sum of the outer product of vectors representing each entity. In this case, the relation is represented by the tensor $T = J \otimes L \otimes M + S \otimes L \otimes T$. The second method assumes additional connections for placing entity vectors onto the appropriate axis of the tensor before computing the outer product. This can be achieved by taking a special outer product of the entity vector with its role vector, for example, $J \otimes' A = (J, -, -)$; $L \otimes' R = (-, L, -)$; and $M \otimes' P = (-, -, M)$, where $-$ indicates no activity passed to that group of units.

In either method, there is an explicit representation of the role vector, which is used to retrieve any relational element by a network of units implementing the inner product. By this formulation, part of the problem of proposing mechanisms for redescription becomes the problem of finding the appropriate entity (filler) and role vectors in complex environmental input. Initial steps towards learning role vectors from tensor representations were done in Phillips (1994, 1995), although more work is required to apply these ideas to mechanisms of redescription.

Conclusion

Relational complexity is not intended as a replacement for representational redescription. It is intended to help constrain and identify possible mechanisms. Relation based capacity limitations posit four major milestones in development: unary to quaternary that are enabled by maturation, but also dependent on experience. Representational redescription is a cyclic process driven both externally and internally. These two positions are not contradictory. Instead, relational complexity places an upper bound on the sorts of internal representational changes performed.

Karmiloff-Smith argued that young children have difficulty on modifications that require interrupting their normal drawing procedures. We have shown that this explanation can be made more formal in terms of a relational model. We suggest that mechanisms for redescription are to be found in models that develop the capacity to represent and process increasingly higher arity relations.

Acknowledgments

We thank the two reviewers for their thoughtful comments, which helped improve the presentation of this work. We also thank members of the Cognitive Development lab at ETL for helpful discussions.

REFERENCES

Andrews, G. (1997). *Relational complexity as a capacity construct in cognitive development.* Ph.D. thesis, School of Psychology, The University of Queensland, Brisbane, Australia.

Boden, M. A. (1994). Representational redescription: A question of sequence. *Behavioral and Brain Sciences*, *17*(4), 708.

Clark, A., & Karmiloff-Smith, A. (1993). The cognizer's innards: A psychological and philosophical pespective on the development of thought. *Mind and Language*, *8*, 487–519.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Halford, G. S. (1993). *Children's understanding: The development of mental models.* Hillsdale, NJ: Lawrence Erlbaum.

Halford, G. S., Wilson, W. H., Guo, J., Gayler, R. W., Wiles, J., & Stewart, J. E. M. (1994). Connectionist implications for processing capacity limitations in analogies. In K. J. Holyoak & J. Barnden (Eds.), *Advances in Connectionist and Neural Network theory*, chap. 7. Norwood, NJ: Ablex.

Halford, G. S., Wilson, W. H., & Phillips, S. (in press). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences.*

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366.

Karmiloff-Smith, A. (1986). From meta-processes to conscious access: Evidence from children's metalinguistic and repair data. *Cognition*, *23*, 95–147.

[6]What Smolensky calls *fillers*.

Karmiloff-Smith, A. (1990). Constraints on representational change: Evidence from children's drawings. *Cognition*, *34*, 57–83.

Karmiloff-Smith, A. (1992). *Beyond Modularity: A developmental perspective on cognitive science.* Cambridge, MA: MIT Press/Bradford Books.

Phillips, S. (1994). Strong systematicity within a connectionist framework: The tensor-recurrent network. In A. Ram & K. Eiselt (Eds.), *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pp. 723–727. Lawrence Erlbaum.

Phillips, S. (1995). *Connectionism and the problem of systematicity.* Ph.D. thesis, The University of Queensland, Department of Computer Science, Brisbane, Australia.

Phillips, S. (1997). Measuring relational complexity in oddity discrimination tasks. *Noetica*, *3*(1).

Phillips, S., Halford, G. S., & Wilson, W. H. (1995). The processing of associations versus the processing of relations and symbols: A systematic comparison. In J. D. Moore & J. F. Lehman (Eds.), *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, pp. 688–691.

Piaget, J. (Ed.). (1957). *Logic and Psychology.* New York, NY: Basic Books.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159–216.

Appendix A: Relations

The preceding arguments were based on standard definitions of sets, relations and their operators. More formal discussion is provided here to help ground these arguments.

A set S is a collection of abstract entities x_i that satisfies property P. That is, $S = \{x_i | P(x_i)\}$. Relation $R^n(A_1, \ldots A_n)$ is a set of tuples $(x_1, \ldots, x_n)$ from the product of sets $S_1, \ldots, S_n$ satisfying property P. That is, $R^n(A_1, \ldots A_n) = \{(x_1, \ldots, x_n) \in S_1 \otimes \ldots \otimes S_n | P(x_1, \ldots, x_n)\}$. Examples of relations are: *even numbers* $R^1_{\text{even}}(X) = \{2, 4, \ldots\}$; *greater than* $R^2_>(X, Y) = \{(3,1), (3,2), \ldots\}$; and *addition* $R^3_+(X, Y, Z) = \{(1,2,3), (1,3,4), \ldots\}$. A nullary relation is just an atomic entity - $R^0_x = x$. So, for example, while the set of even numbers is a unary relation, its elements 2, 4, etc are constant, atomic entities and may be considered nullary relations.

As mathematical objects, there are infinitely many possible relations, but not all of these can be stored in a database, or represented in a cognitive system. Most of these relations are derivable via combinations of relational operators. Basic relational operators include: *select* and *project* for retrieving a relation from within a relation (e.g., retrieve only those numbers less than two); and *join* for constructing larger relations from smaller ones. Details of these operators can be found in Phillips et al. (1995).

A binary relation can be constructed by taking the *outer join* of two unary relations. An outer join concatenates every tuple in the first relation with every tuple in the second relation. An *equi-join* concatenates only those tuples that share common elements along the specified dimensions of the two relations. Equi-joins can be used to make transitive inferences. Suppose the binary relation $R^2_<(X, Y) = \{(1,2), (2,3), (3,4), (4,5)\}$. Joining the relation with itself at the Y dimension in the first instance, and the X dimension in the second instance returns $R^3_<(X, Y{=}X, Y) = \{(1,2,3), (2,3,4), (3,4,5)\}$. Projecting onto the first and third dimensions (positions) results in $R^2_<(X, Y) = \{(1,3), (2,4), (3,5)\}$.

Common knowledge of relations comes from their use in language. But, relational complexity is not simply a count of the nouns in a sentence. It involves an analysis of the functional dependencies between relation elements. Suppose the following facts:

- *There is an apple and an orange on the table.*
- *There is a jacket and a hat on the chair.*

At first glance, these two facts appear to involve a ternary relation between two objects and their support: $R^3_{\text{on}}(O_1, O_2, S) = \{(a, o, t), (j, h, c)\}$. But, the same information is represented by the binary relation: $R^2_{\text{on}}(O, S) = \{(a, t), (o, t), (j, c), (h, c)\}$, since the two facts can be reconstructed by joining the binary relation with itself along the support dimension (S). Determining, for example, where is the jacket requires only processing a binary relation.