Toward a Soar Theory of Taking Instructions for Immediate Reasoning Tasks

Richard L. Lewis, Allen Newell, and Thad A. Polk
School of Computer Science
Carnegie Mellon University

(Richard.Lewis@cs.cmu.edu)

March 31, 1989

Soar is a theory of human cognition [New89], embodied in a computer system. Soar specifies the cognitive architecture, which is the relatively fixed set of mechanisms that permit goals and knowledge of the task environment to be encoded in memory and brought to bear to produce behavior. Soar is proposed as a unified theory of cognition, and it has been applied to human behavior in a broad spectrum of domains. This paper reports progress in getting Soar to take instructions and organize itself to perform a required task.

There are three important reasons for wanting a theory of instructions. First, taking instructions is a domain of cognitive activity, with interesting phenomena and practical importance. Any unified theory of cognition must ultimately provide such a theory. Second, a major issue for psychology has always been the radical underdetermination of theory by data. Though an issue for all sciences, it is particularly irksome for psychology (and the social sciences) because humans bring massive knowledge to a task and dynamically organize themselves accordingly. Instructions to perform tasks are an important instance of such self-organization (e.g., as it takes place in psychological experiments). A theory of how humans take instructions would go some way toward removing what can be called *theory degrees of freedom*. Instructions are only one source of knowledge that determines behavior, but success in dealing with them could point toward how to tackle other sources. Third, within a unified theory of cognition, including both taking instructions and performing the subsequent task in the same theoretical account would provide mutual constraint. This would provide an instance of the gains purportedly to be made from unification.

In this paper we take some steps toward this goal[1]. The system we present here, NL-Soar, comprehends instructions given in elementary English for *immediate reasoning* tasks, such as the *relational reasoning* task [Joh88] shown in Figure 1. This comprehension is part of a system, IR-Soar, that models the way humans perform immediate reasoning (reported in a companion paper [PNL89]). Here we focus on the internal representation of instructions, how Soar organizes itself to do the task, and the associated psychological claims. The process of comprehending the language of instructions to create these representations is also part of the theory of cognition, and involves issues both of linguistics and psycholinguistics. Although we do not deal here with these issues, NL-Soar embodies a theory of language comprehension [New89, Chap. 8]. We also do not present direct behavioral evidence for instruction taking. For the moment, the psychological relevance of the instruction taking is that it leads to an organization of IR-Soar that explains how people do these tasks.

There has not been much work on the psychological theory of instruction taking. The most notable was the UNDERSTAND program [SH79], which took instructions for the Tower of Hanoi and constructed a problem space in which to do the task. Our account is consonant with the broad thrust of that work, the

---

[1] Building on earlier work in [New89, Chap. 7, YN88].

Instructions

Read four premises.
Then read a statement.
If the statement is true say "true".
Then stop.

Task input

Premises	A plate is left of a knife. A fork is left of the plate. A jug is above the knife. The fork is below a cup.
Statement	The cup is left of the jug.

Figure 1: Relational reasoning task.

main advances being in the plausibility of the processes and representations used, and in the embedding of this in a unified theory. Our account is also consonant with the implications from ACT* [And83]: that conversion from declarative to procedural form occurs by an interpretive process that leads to chunking filaments of conditional behavior (productions).

We first present the psychological claims of the Soar theory of taking instructions, and in passing review the basic assumptions of the Soar architecture. We then illustrate the theory by tracing the behavior of the system in detail on the relational reasoning task in Figure 1. Finally, we briefly describe how the theory has been applied to two other immediate reasoning tasks.

The Soar Theory of Taking Instructions

Soar as a cognitive architecture has been described in several places [LNR87] and we will take its major outline to be familiar. All tasks are formulated in *problem spaces*; all long term knowledge is held in a *recognition memory* (realized as productions); processing proceeds by a sequence of *decision cycles* that accumulate knowledge about what spaces, states and operators to select; *subgoals* are generated in an attempt to resolve *impasses* that occur when the decision-making knowledge is insufficient or conflicting; and the experience gained in resolving impasses is learned in the form of *chunks* (new productions in recognition memory).

One additional assumption is that states in problem spaces are *annotated models*.[2] Models consist of objects, properties, and relations (model elements) and satisfy the semantic assumption that each element in the representation corresponds to an element in the referent. This assumption may be violated in principled ways by attaching annotations to model elements. An annotation specifies a non-standard interpretation for the single element to which it is attached (e.g., an element annotated **many** corresponds to multiple elements in the referent). There are computational advantages to processing annotated models and there is also evidence that humans use them [Joh83,PN88]. We do not review these considerations, but simply assume annotated models for all representations.

Beyond these fundamental assumptions, the Soar theory of taking instructions makes the following psychological claims (elaborated below):

---

[2] This is not yet an architectural assumption for Soar per se, which only assumes a representation consisting of attributes and values. Nevertheless, all the current work in Soar on human cognition assumes models [New89, Chap. 7], [PN88].

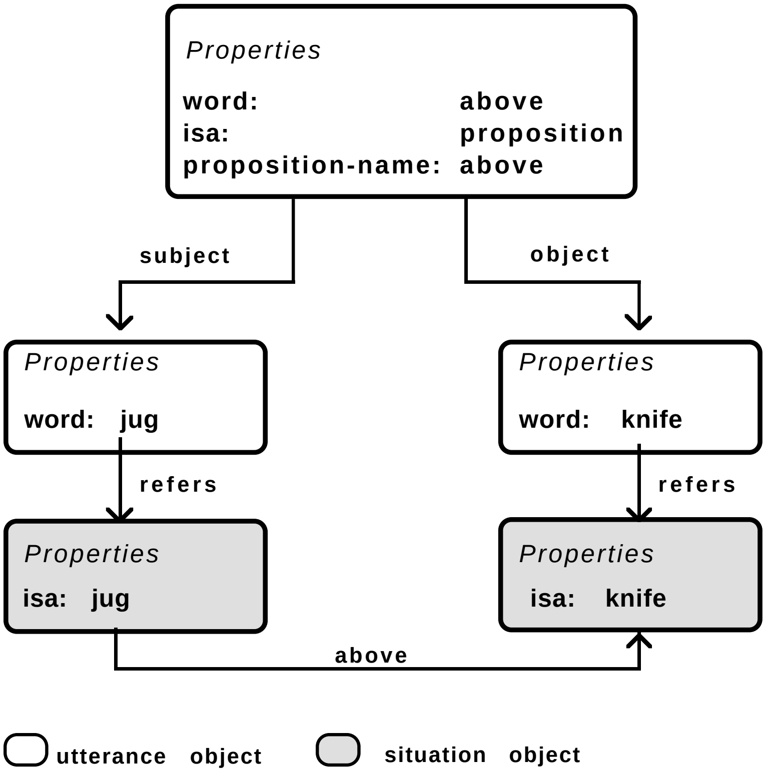

Figure 2: Utterance and situation model for "A jug is above the knife."

1. **Situation model.** The objective of comprehension is to represent the situation being described. To do so, comprehension builds a *model of the situation*.
2. **Utterance model.** As a processing side effect, comprehension produces a *model of the utterance*—a processed linguistic form that can be interpreted as an abstract proposition.
3. **Behavior model.** If instructions are being comprehended, then comprehension produces a *model of the subject's future behavior.*
4. **Performance by interpretation.** Initially, task performance proceeds by *interpretively executing* the behavior model. During this interpretation process, chunks are learned that directly perform the task.

To comprehend an utterance means to have an annotated model representing the situation described by that utterance. In NL-Soar, comprehension occurs in the **comprehend** problem space by applying a *comprehension operator* to each incoming word [New89]. This space iteratively augments and refines the situation model based on the input. Figure 2, bottom, shows a simple situation model.

A large part of the processing in **comprehend** involves integrating new information into a partial model in an appropriate way. This requires *expectation* data structures that hold partial comprehension knowledge. A set of match problem spaces is used to link up expectations with the model elements to which they correspond. These spaces compose a powerful recursive match out of a basic attention mechanism (described in [PNL89]).

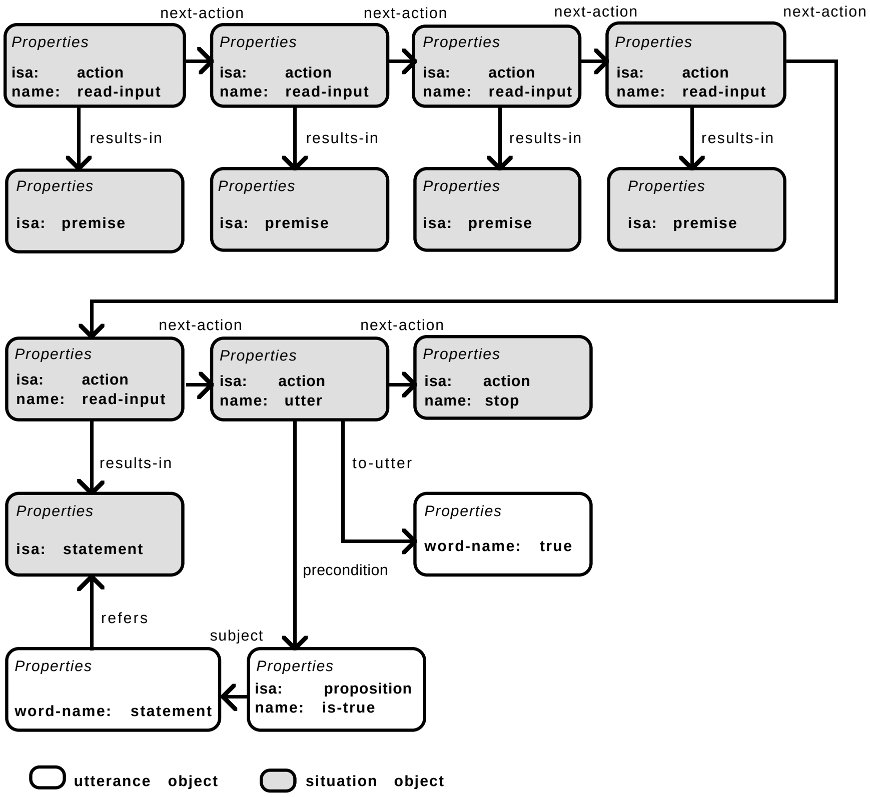

Figure 3: Behavior model for the relational reasoning task.

Expectations can be semantic, syntactic, and pragmatic; hence, there must be some way of modeling the linguistic structure of the sentence. The *utterance model* serves this processing requirement. The utterance model begins as a simple string of words, but as comprehension proceeds, it evolves into a model that reflects the *propositional content* of the utterance and becomes closely linked to the situation model. Figure 2, top, gives the final utterance model for "A jug is above the knife." The model consists of additional structure imposed on the words in the utterance (e.g., the relations *subject* and *object*). This structure can be interpreted as the abstract proposition or description that is asserted by the utterance. Also note that parts of the utterance model are linked to their corresponding situation model parts. The production of the utterance model is a side effect of the comprehension process, but it is a requirement for dealing with language. It is not a deliberate product of comprehension. The objective of comprehension is to represent the situation that the utterance designates. Not everything can necessarily be encoded in the situation model, so the total knowledge provided by an utterance may be distributed between the two models.

The result of comprehending instructions is to produce a behavior model that represents future actions to be taken. Figure 3 shows the behavior model that NL-Soar produces for the relational reasoning task in Figure 1. A part in the behavior model corresponds to an action or an object related to an action (such as the expected input or output). For example, the model in Figure 3 specifies that the task begins with four acts of reading input, and that each such act should yield a premise.

After reading instructions, the system attempts to perform the task. It is initially unable to proceed, because it lacks operational knowledge of the task in recognition memory (that knowledge is in the model of the desired future behavior). This leads to interpretation of the behavior model. Earlier work has shown how such interpretive processes can lead to chunks that directly implement the task [YN88,New89]. This interpretive behavior is embodied in a set of problem spaces that are independent of the others in NL-Soar and those in IR-Soar. For immediate reasoning tasks, the interpretation of the behavior model gives rise to task operators that use the three basic IR-Soar problem spaces (**comprehend**, **test-proposition**, and **build-proposition**) to perform these tasks. Polk, Newell, and Lewis (1989) show that the problem spaces so acquired can indeed be used to model human performance on such tasks.

An Example: Relational Reasoning

We illustrate the theory by tracing through the task of Figure 1. Figure 4 shows the behavior of the system as it comprehends the instructions and attempts to perform the task. The system begins in the **read** problem space, and applies a series of **comprehend-input** operators to read the instructions. Each operator application comprehends one statement, and builds up the behavior model accordingly (the ovals). The **comprehend-input** operator is implemented in the **comprehend** space, where a series of comprehension operators fire for each incoming word. (They actually fire multiple times as the expectations build up and are satisfied by incoming words and comprehended pieces of the utterance model.)

Processing continues in this fashion until comprehension of the word "begin," that the system takes to mean it should begin performing the task. It does this by selecting the operator do-new-task, and creating a new problem space (**relation**) to implement the operator.

At this point, Soar impasses because it has no operators to propose for the new space. Resolving the impasse requires consulting the behavior model to determine what to do next. This is the function of the **fetch-operator** space, a problem space, that contains the knowledge required to locate the next action in the behavior model and instantiate it as an operator in the task space. In this case, the impasse is resolved by selecting an instantiation of the **comprehend-input** operator that will yield a premise (Figure 3). The premise object from the behavior model is set up as an expectation in the **comprehend** space, and thus provides the goal test for the **comprehend-input** operator.

This *impasse-fetch-apply* cycle continues until Soar interprets the utter action in the behavior model. This action has a precondition, represented as a proposition (Figure 3) that originated as part of the utterance model for one of the comprehended instructions. Determining if this action should be taken requires verifying the proposition, so the impasse in the **relation** space is resolved by selecting the **test-prop** operator. This operator is implemented in in the **test-proposition** space, one of the three basic IR-Soar problem spaces [PNL89]. Once the proposition has been verified, the **fetch-operator** space instantiates and selects the **utter** operator in the next fetch cycle. Finally, the **stop-task** operator is selected and terminates the task.

Other Tasks

NL-Soar has acquired two other immediate reasoning tasks. Figure 5 shows the instructions and problem-spaces for the Clark and Chase sentence verification task [CC72], and the categorical syllogisms task. As in the relational reasoning task, task-specific behavior arises by interpreting the behavior model, and applying operators in the new task space that are implemented in IR-Soar's **comprehend**, **test-proposition**,

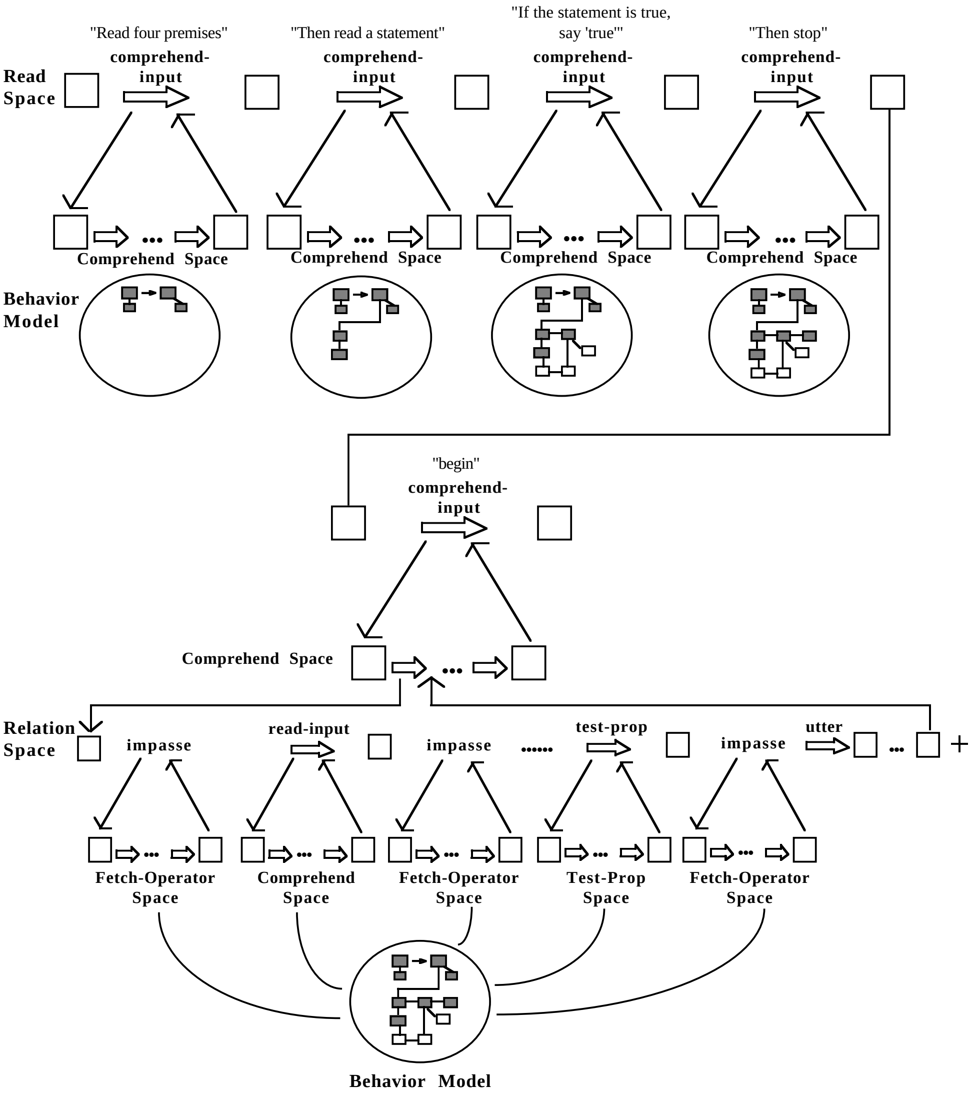

Figure 4: Acquiring and performing the relational reasoning task.

Categorical Syllogisms

Instructions	Syllogism Problem Space
1. Read two premises that share a term.	1. Read-input [comprehend]
2. Then produce a statement that follows from the premises.	2. Make-conclusion [build-proposition]
3. The statement relates the unique terms of the premises.	3. Make-conclusion [goal-test]
4. Then stop.	

Clark and Chase

Instructions	C & C Problem Space
1. Examine the picture.	1. Read-input [comprehend]
2. Then read the statement.	2. Read-input [comprehend]
3. If the statement is true press the t-button.	3. Test-prop [test-proposition]
4. If the statement is false press the f-button.	4. Test-prop [test-proposition]
5. Then stop.	

Figure 5: Other immediate reasoning tasks.

or **build-proposition** problem spaces[3].

The Clark and Chase task differs from relational reasoning in the simplicity of the initial situation, and the form used to present the situation (a picture). The latter difference shows up in the behavior model as a **comprehend-input** action that expects a picture rather than a linguistic utterance. The difference in simplicity is a function of the task input, not the task instructions.

The syllogisms task [PNL89] is interesting because the subject must utter a conclusion that conforms to a particular specification given in the instructions— namely, that the conclusion relate the *unique terms of the premises.* Soar realizes this as an application of the **build-proposition** operator instantiated to relate the correct terms. Knowledge in the comprehension operators for the words "unique" and "relate" leads to construction of the appropriate behavior model that encodes this constraint.

Conclusion

We have presented a Soar theory of taking instructions for immediate reasoning tasks. This theory is implemented in a Soar system, NL-Soar. NL-Soar uses the **comprehend** problem space to read simple English statements and produce an annotated model of the situation being described. As a side effect of comprehending these statements, **comprehend** produces a model that reflects the linguistic structure of the utterance. When reading task instructions, comprehension creates a model of the behavior described by the instructions. By repeatedly consulting this behavior model, the system can acquire the problem spaces necessary to perform the task.

Besides being interesting in its own right, this theory opens up some interesting possibilities. For one, it significantly alleviates the problem of the underdetermination of theories by data. In a companion paper [PNL89], we have presented a theory of immediate reasoning that depends on the task-specific problem spaces that arise from the instructions given to NL-Soar. The degrees of freedom available to that theory are significantly reduced as a result. Further, the two subtheories of taking instructions and immediate reasoning mutually constrain each other, making both significantly stronger. For instance, it is not an independent assumption of the immediate reasoning theory that both the proposition model

---

[3] NL-Soar will deal with the conditional reasoning and Wason tasks, which are examples in [PNL89]; as of submission of this draft, the runs are not completed, but no difficulties are expected.)

and the situation model are available as sources of knowledge to do the task. Similarly, the problem spaces acquired through task instructions must be used to model behavior in immediate reasoning tasks, significantly constraining the theory presented here. Finally, this theory represents another step toward making Soar a unified theory of cognition.

Acknowledgements

We would like to thank Erik Altmann, Norma Pribadi, Kathryn Swedlow, and Shirley Tessler for useful comments and invaluable assistance in preparing the draft and the figures, as well as Chris Tuck for helping bridge the Pittsburgh-Ann Arbor gap. This work was supported by the Information Sciences Division, Office of Naval Research, under contract N00014-86-K-0678, and by the National Science Foundation and Kodak fellowship programs in which Richard Lewis and Thad Polk, respectively, participate. The views expressed in this paper are those of the authors and do not necessarily reflect those of the supporting agencies. Reproduction in whole or in part is permitted for any purpose of the United States government. Approved for public release; distribution unlimited.

References

[And83] J. R. Anderson. *The Architecture of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1983.

[CC72] H.H. Clark and W.G. Chase. On the process of comparing sentences against pictures. *Cognitive Psychology*, 3:472–517, 1972.

[Joh83] P. N. Johnson-Laird. *Mental models: Towards a cognitive science of language, inference and consciousness*. Harvard University Press, Cambridge, Massachusetts, 1983.

[Joh88] P.N. Johnson-Laird. Reasoning by rule or model? In *Proceedings of the Annual Conference of the Cognitive Science Society*, pages 765–771, August 1988.

[LNR87] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[New89] A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1989. In press.

[PN88] T. A. Polk and A. Newell. Modeling human syllogistic reasoning in Soar. In *Proceedings of the Annual Conference of the Cognitive Science Society*, pages 181–187, August 1988.

[PNL89] T.A. Polk, A. Newell, and R.L. Lewis. Toward a Unified Theory of Immediate Reasoning in Soar. March 1989. School of Computer Science, Carnegie Mellon University, Unpublished.

[SH79] H.A. Simon and J.R. Hayes. Understanding written problem instructions. In H.A. Simon, editor, *Models of Thought*, pages 451–476, Yale University Press, New Haven, Conneticut, 1979.

[YN88] G. R. Yost and A. Newell. Learning New Tasks in Soar. February 1988. Unpublished.