ON THE EXISTENCE OF PRIMITIVE MEANING UNITS

Sharon C. Salveter
Computer Science Department
SUNY Stony Brook
Stony Brook, N.Y. 11794

ABSTRACT

Knowledge representation schemes are either based on a set of primitives or not. The decision of whether or not to have a primitive-based scheme is crucial since it affects the knowledge that is stored and how that knowledge may be processed. We suggest that a knowledge representation scheme may not initially have primitives, but may evolve into a primitive-based scheme by inferring a set of primitive meaning units based on previous experience. We describe a program that infers its own primitive set and discuss how the inferred primitives may affect the organization of existing information and the subsequent incorporation of new information.

1. DECIDING HOW TO REPRESENT KNOWLEDGE

A crucial decision in the design of a knowledge representation is whether to base it on primitives. A primitive-based scheme postulates a pre-defined set of meaning structures, combination rules and procedures. The primitives may combine according to the rules into more complex representational structures, the procedures interpret what those structures mean. A primitive-free scheme, on the other hand, does not build complex structures from standard building blocks; instead, information is gathered from any available source, such as input and information in previously built meaning structures.

A hybrid approach postulates a small set of pre-defined meaning units that may be used if applicable and convenient, but is not limited to those units. Such a representation scheme is not truly primitive-based since the word "primitive" implies a complete set of pre-defined meaning units that are the only ones available for construction. However, we will call this hybrid approach a primitive-based scheme, since it does postulate some pre-defined meaning units that are used in the same manner as primitives.

2. WHAT IS A PRIMITIVE?

All representation systems must have primitives of some sort, and we can see different types of primitives at different levels. Some primitives are purely structural and have little inherent associated semantics. That is, the primitives are at such a low level that there are no semantics pre-defined for the primitives other than how they may combine. We call these primitives structural primitives. On the other hand, semantic primitives have both structural and semantic components. The structures are defined on a higher level and come with pre-attached procedures (their semantics) that indicate what they "mean," that is, how they are to be meaningfully processed. What makes primitives semantic is this association of procedures with structures, since the procedures operating on the structures give them meaning. In a primitive-based scheme, we design both a set of structures and their semantics to describe a specific environment.

There are two problems with pre-defining primitives. First, the choice of primitives may be structurally inadequate. That is, they may limit what can be represented. For example, if we have a set of rectilinear primitives, it is difficult to represent objects in a sphere world. The second problem may arise even if we have a structurally adequate set of primitives. In this case the primitives may be defined on too low a level to be useful. For example, we may define atoms as our primitives and specify how atoms interact as their semantics. Now we may adequately describe a rubber ball structurally, but we will have great difficulty describing the action of a rolling ball. We would like a set of semantic primitives at a level both structurally and semantically appropriate to the world we are describing.

3. INFERRING AN APPROPRIATE PRIMITIVE SET

Schank [1972] has proposed a powerful primitive-based knowledge representation scheme called conceptual dependency. Several natural language understanding programs have been written that use conceptual dependency as their underlying method of knowledge representation. These programs are among the most successful at natural language understanding. Although Schank does not claim that his primitives constitute the only possible set, he does claim that some set of primitives is necessary in a general knowledge representation scheme.

Our claim is that any advanced, sophisticated or rich memory is likely to be decomposable into primitives, since they seem to be a reasonable and efficient method for storing knowledge. However, this set of after-the-fact primitives need not be pre-defined or innate to a representation scheme; the primitives may be learned and therefore vary depending on early experiences.

We really have two problems: inferring from early experiences a set of structural primitives at an appropriate descriptive level and learning the semantics to associate with these structural primitives. In this paper we shall only address the first problem. Even though we will not address the semantics attachment task, we will describe a method that yields the minimal structural units with which we will want to associate semantics. We feel that since the inferred structural primitives will be appropriate for describing a particular environment, they will have appropriate semantics and that unlike pre-defined primitives, these learned primitives are guaranteed to be at the appropriate level for a given descriptive task. Identifying the structural primitives is the first step (probably a parallel step) in identifying semantic primitives, which are composed of structural units and associated procedures that give the structures meaning.

This thesis developed while investigating learning strategies. Moran [Salveter 1979] is a program that learns frame-like structures that represent verb meanings. We chose a simple representative frame-like knowledge representation for Moran to learn. We chose a primitive-free scheme in order not to determine the level of detail at which the world must be described.

As Moran learned, its knowledge base, the verb world, evolved from nothing to a rich interconnection of frame structures that represent various senses of different root verbs. When the verb world was "rich enough" (a heuristic decision), Moran detected substructures, which we call building blocks, that were frequently used in the representations of many verb senses across root verb boundaries. These building blocks can be used as after-the-fact primitives. The knowledge representation scheme thus evolves from a primitive-free state to a hybrid state. Importantly, the building blocks are at the level of description appropriate

to how the world was described to Moran. Now Moran may reorganize the interconnected frames that make up the verb world with respect to the building blocks. This reorganization results in a uniform identification of the commonalities and differences of the various meanings of different root verbs. As learning continues the new knowledge incorporated into the verb world will also be stored, as much as possible, with respect to the building blocks; when processing subsequent input, Moran first tries to use a combination of the building blocks to represent the meaning of each new situation it encounters.

A set of building blocks, once inferred, need not be fixed forever; the search for more building blocks may continue as the knowledge base becomes richer. A different, "better," set of building blocks may be inferred later from the richer knowledge and all knowledge reorganized with respect to them. If we can assume that initial inputs are representative of future inputs, subsequent processing will approach that of primitive-based systems.

4. AN OVERVIEW OF MORAN

Moran is able to "view" a world that is a room; the room contains people and objects. Moran has pre-defined knowledge of the contents of the room. For example, it knows that lamps, tables and chairs are all types of furniture, Figaro is a male, Ristin is a female, Ristin and Figaro are human. As input to a learning trial, Moran is presented with:

1) a snapshot of the room just before an action occurs,

2) a snapshot of the room just after the action is completed and

3) a parsed sentence that describes the action that occured in the two-snapshot sequence.

The learning task is to associate a frame-like structure, called a Conceptual Meaning Structure (CMS), with each root verb it encounters. A CMS is a directed acyclic graph that represents the types of entities that participate in an action and the changes the entities undergo during the action.

The CMSs are organized so that the similarities among various senses of a given root verb are explicitly represented by sharing nodes in a graph. A CMS is organized into two parts: an arguments graph and an effects graph. The arguments graph stores cases and case slot restrictions, the effects graph stores a description of what happens to the entities described in the arguments graph when an action "takes place."

A simplified example of a possible CMS for the verb "throw" is shown in Figure 1. Sense 1, composed of argument and effect nodes labelled A, W and X can represent "Mary throws the ball." It show that during sense 1 of the action "throw," a human agent remains at a location while a physical object changes location from where the Agent is to another location. The Agent changes from being in a state of physical contact with the Object to not being in physical contact with it. Sense 2 is composed of nodes labelled A, B, W and Y; it might represent "Figaro throws the ball to Ristin." Sense 3, composed of nodes labelled A, B, C, W, X and Z, could represent "Sharon threw the terminal at Raphael."

Moran infers a CMS for each root verb it encounters. Although similarities among different senses of the same root verb are recognized, similarities are not recognized across CMS boundaries; true synonyms might have identical graphs, but Moran would have no knowledge
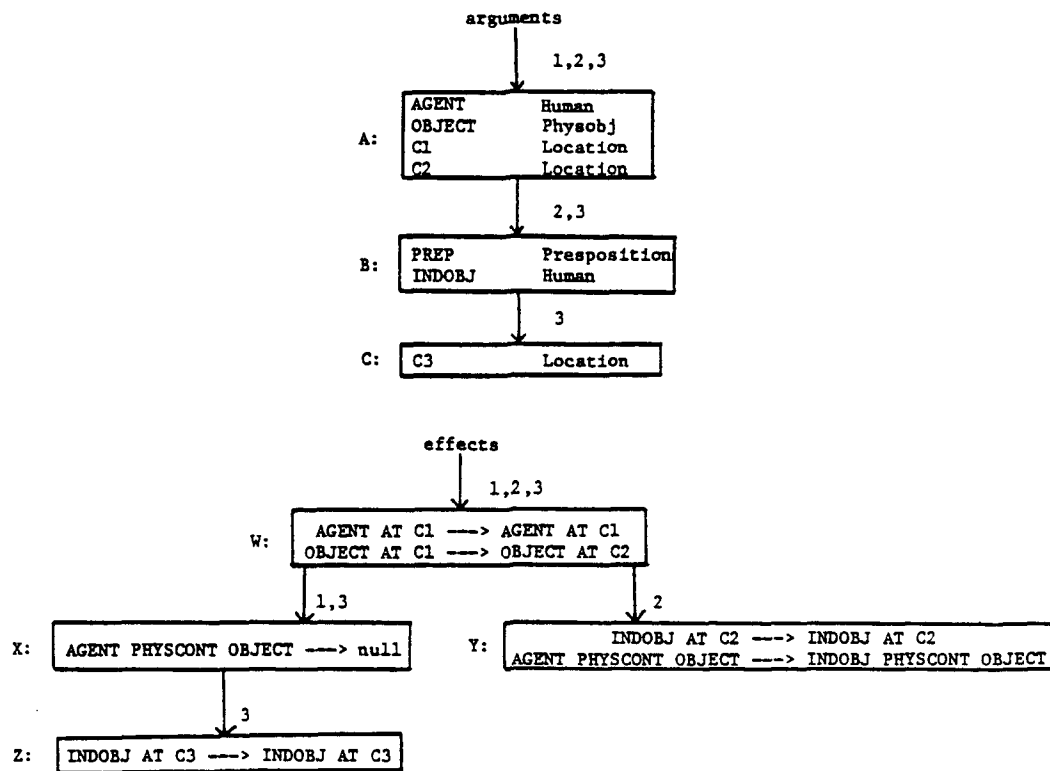

Figure 1.

of the similarity. Similarities among verbs that are close in meaning, but not synonyms, are not represented; the fact that "move" and "throw" are related is not obvious to Moran.

5. PRELIMINARY RESULTS

A primitive meaning unit, or building block, should be useful for describing a large number of different meanings. Moran attempts to identify those structures that have been useful descriptors. At a certain point in the learning process, currently arbitrarily chosen by the human trainer, Moran looks for building blocks that have been used to describe a number of different root verbs. This search for building blocks crosses CMS boundaries and occurs only when memory is rich enough for some global decisions to be made.

Moran was presented with twenty senses of four root verbs: move, throw, carry and buy. Moran chose the following effects as building blocks:

1) Agent (human) AT Case1 (location)
   ↓
   Agent (human) AT Case1 (location)
   * a human agent remains at a location *

2) Agent (human) AT Case1 (location)
   ↓
   Agent (human) AT Case2 (location)
   * a human agent changes location *

3) Object (physicalobj) AT Case1 (location)
   ↓
   Object (physicalobj) AT Case2 (location)
   * a physical object changes location *

4) Agent (human) PHYSICALCONTACT Object (physicalobj)
   ↓
   Agent (human) PHYSICALCONTACT Object (physicalobj)
   * a human agent remains in physical contact with a physical object *

Since Moran has only been presented with a small number of verbs of movement, it is not surprising that the building blocks it chooses describe Agents and Objects moving about the environment and their interaction with each other. A possible criticism is that the chosen building blocks are artifacts of the particular descriptions that were given to Moran. We feel this is an advantage rather than a drawback, since Moran must assume that the world is described to it on a level that will be appropriate for subsequent processing.

In Schank's conceptual dependency scheme, verbs of movement are often described with PTRANS and PROPEL. It is interesting that some of the building blocks Moran inferred seem to be subparts of the structures of PTRANS and PROPEL. For example, the conceptual dependency for "X throw Z at Y" is:

$$X \Longleftrightarrow \text{PROPEL} \xleftarrow{o} Z \xleftarrow{D} \begin{array}{c} \longrightarrow Y \\ \longleftarrow X \end{array}$$
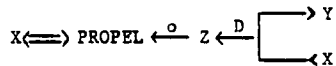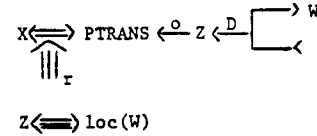
where X and Y are humans and Z is a physical object. We see the object, Z, changing from the location of X to that of Y. Thus, the conceptual dependency subpart:

$$\xleftarrow{o} Z \xleftarrow{D} \begin{array}{c} \longrightarrow \\ \longleftarrow \end{array}$$
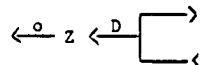
appears to be approximated by building block #3 where the Object changes location. Moran would recognize that the location change is from the location of the

Agent to the location of the indirect object by the interaction of building block #3 with other building blocks and effects that participate in the action description.

Similarly, the conceptual dependency for "X move Z to W" is:

$$X \Longleftrightarrow \text{PTRANS} \xleftarrow{o} Z \xleftarrow{D} \begin{array}{c} \longrightarrow W \\ \longleftarrow \end{array}$$
$$\underset{r}{\text{|||}}$$

$$Z \Longleftrightarrow \text{loc}(W)$$

where X and Z have the same restrictions as above and W is a location. Again we see an object changing location; a common occurence in movement and a building block Moran identified.

6. CONCLUDING REMARKS

We are currently modifying Moran so that the identified building blocks are used to process subsequent input. That is, as new situations are encountered, Moran will try to describe them as much as possible in terms of the building blocks. It will be interesting to see how these descriptions differ from the ones Moran would have constructed if the building blocks had not been available. We shall also investigate how the existence of the building blocks affects processing time.

As a cognitive model, inferred primitives may account for the effects of "bad teaching," that is, an unfortunate sequence of examples of a new concept. If examples are so disparate that few building blocks exist, or so unrepresentative that the derived building blocks are useless for future inputs, then the after-the-fact primitives will impede efficient representation. The knowledge organization will not tie together what we have experienced in the past or predict that we will experience in the future. Although the learning program could infer more useful building blocks at a later time, that process is expensive, time-consuming and may be unable to replace information lost because of poor building blocks chosen earlier. In general, however, we must assume that our world is described at a level appropriate to how we must process it. If that is the case, then inferring a set of primitives is an advantageous strategy.

REFERENCES

[Salveter 1979] Inferring conceptual graphs. *Cognitive Science*, 1979, 3, 141-166.

[Schank 1972] Conceptual Dependency: a theory of natural language understanding. *Cognitive Psychology*, 1972, 3, 552-631.