L_0 Terminology

J. Feldman, D. Bailey, S. Narayanan, J. Segal, etc.

June 7, 1995

This document serve to define various terms used within the L_0 project. It contains a discussion of the rationale behind many of the terms, a top-level diagram of the current L_0 architecture, and a comprehensive glossary.

SO FAR IT'S JUST A DRAFT!

Discussion

Motivation

The current L_0 tasks of learning and understand action-oriented language involve the integrate a wide range of capabilities. An assumption we make is that we should not try to make do with a single representation and processing mechanism. To minimize chaos we organize our various representations and processes into **components**, and use specialized terminology within each. ("Module" would be a natural computer-science term but would have unwanted connotations to a psychologist.) We distinguish a component's **function** from its **realization**.

Overview

The main new ideas focus on components which are **executing**. Executing components integrate representation with processing. Put another way, executing components may be actual controllers of activity, or mental simulators of activity or of the world. We think they are the most natural way to reason about processes. An alternative term might be "active representations". "Procedural representations" is close, but fails to convey that (1) the content comes from the actual running of the procedure, as opposed to some outside entity inspecting the (passive) code; and (2) the representation's nature is greatly affected by the fact that it *is* the computational substrate, not merely "code" which assumes some outside substrate upon which to run.

However not all components are executing. Components which involve passive, "declarative" representations which are operated upon by separate processing mechanisms, are called **descriptive** components. In particular we focus on **feature structures** (**f-structures** for short), which are lists of **feature–value** pairs. In other words, a "slot-and-filler" representation (though beware the term "slot" is used elsewhere). Values may be discrete or

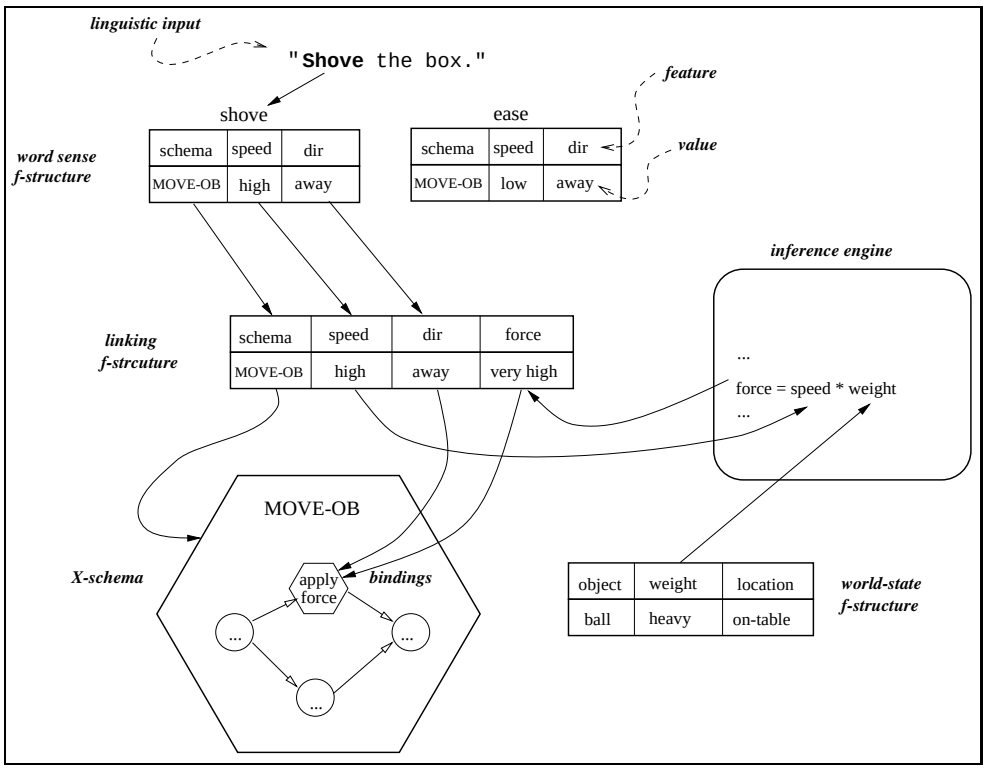

Figure 1: L_0 componential structure with terminology in italics.

continuous, and in some places we may use probability distributions over the possible values for a feature. This usage of the term "feature" is intended to encompass the linguistic notion of the same name, although the actual features we will work with are likely to be more numerous, detailed, and "bodily-grounded" than is seen in some linguistic analyses.

Our f-structures are also compatible with standard symbolic AI role/filler structures, which leads to an important advantage, namely that we can easily imagine integrating a wide variety of logical and probabilistic **inference engines** (such as unification, Bayesian inference, etc.) into our language processing system as they become necessary.

Our central claim, of course, is that both executing and descriptive components will be necessary for any solution to language understanding and acquisition, but we would be delighted to learn of purely descriptive approaches.

Basic model components

All of the current work focusses on mapping language to and from bodily actions. A prototypical case is David Bailey's research on learning the verbs (or verb compounds) that best label simple actions of one hand in a simulated blocks world. As always in L_0 the model

should be able to learn any language without changes. Figure 1 diagrams the components involved in this task, and we will walk through the figure here.

Our convention is that f-structures appear as rows or columns of pairs of rectangular boxes. Within a pair, one box displays the feature name and the other displays the current value (or distribution over possible values).

Executing structures are represented as hexagons. Our current formalism for the internal structure of executing components is a modified form of Petri nets extended to be hierarchical and **parameterized**. We may get by with a somewhat simpler representation involving finite state machines with extensions for concurrency, but Petri nets are currently favored for their ease of representing asynchronous event-based control. In any case, structures of these forms for representing particular bodily actions will be called **X-schemas**, or simply **schemas** when (if ever!) no confusion can arise. This term is explicitly intended to evoke the idea of motor schemata and is consistent with the notion of image schemata from cognitive linguistics. Names of X-schemas will be written in all-caps, e.g., MOVE-OBJECT. We depict X-schemas using standard Petri-net or finite-state-machine notation, except that action states which invoke sub-schemas are drawn hexagonallly.

Three particular types of f-structures are utilized in the current model. Each word (or morpheme more generally) is represented by at least one **word-sense f-structure** (or **word-sense** for short), which contains its semantics. The word-sense contains only that subset of the full set of features used in the model which are selected for by the word. The feature values may be represented as distributions, or as a single value with some kind of confidence.

The model also contains a single, special f-structure called the **linking f-structure**, which contains all linguistically-relevant features. This structure is utilized during both labelling of actions, and execution of verbal commands. Its purpose is to convey information between the descriptive and executing components of the model.

Lastly, there is a **world-state f-structure** which contains (potentially dynamic) information about the state of the world and the agent's intentions which is *not* linguistically relevant, but which is needed by the schemas during execution.

Look at the how the model would repond to the command "Shove the box." First, we assume that other components parse the phrase, assign grammatical roles to its constitutents, and identify the objects mentioned. Intuitively, the basic problem is to select the MOVE-OB schema and appropriately set its `force` and `direction` parameters. In general parameters depends on both the (linguistic) features of the utterance and on the (physical) features of the object. For example, the linguistic feature `speed` (pretend to believe that is the correct linguistic feature for the moment) is combined with the world-state feature `weight` to derive the setting for parameter `force`. If "shove" has been learned correctly, the result will be a very high setting for the `force` parameter.

Glossary

binding: An association between a parameter and a value coming from somewhere else, i.e., from linguistic input.

bound schema (instance?): A schema plus bindings for its parameters, i.e., a schema that is ready to run in some world state.

component: Portion of the full system whose **function** is to solve a particular subtask, and whose **realization** is tailored to that subtask and hence may differ from that of other components.

construction: An f-structure representing the semantics of a multi-word utterance. Generally used when the semantics are non-compositional (e.g. idioms).

descriptive: Representations which are passive, i.e., are independent of the processes which operated upon them. In our systems, they contain a subset of the information needed to carry out an action.

executing: Components which contribute to semantics by active processing.

feature: An element of descriptive representation. Meant to include the standard features in linguistics, plus other, more bodily-grounded features.

f-structure: A collection of feature–value pairs.

image schema: Geometric or spatial knowledge represented in a non-symbolic form, *a la* Regier's thesis. May still be considered executing by virtue of its ability to work "in reverse".

inference engine: A process, plus a database of world knowledge, which can inspect, set, or change the values in the linking f-structure, before or during schema execution.

linking f-structure: A special f-structure which is used to interface descriptive-representation-based processes such as language comprehension or inference, to executing representations, such as schemas.

parameter: A "knob" on a schema which allows the schema to be tailored to various cimumstances.

slot: A position within linguistic input, corresponding to a "grammatical role" such as prefix, verb root, or participle. Also refers to machinery used to process input occuring in a given slot.

value: A particular setting for a feature. Or, potentially, a probability distribution over the possible values for a feature.

word sense: An f-structure, probably with probabilistic values, representing the semantics of a word. A word may have several word senses.

world-state f-structure: A special f-structure which holds context needed for schema execution but not for language.

x-schema: An executing component which encodes knowledge of, and indeed controls, motor activity for simple activities.