SMLs - Very Limited Languages or Small Model Languages

Basic Idea

Languages are vast and complex systems. One way of exploring the requirements for a human communication system might be to devise small-scale systems which can create a virtual world of potential messages. What properties should such a small-scale system have? What can small-scale systems do? Do real languages have the predicted properties? Let us call such thought experiments "small model languages" (SMLs).

What do we mean by "model"?

A model (or self-contained set, or relational system) is in its simplest form a pair, consisting of a set of names of entities and the names of relations between those entities.

$$\textit{Model} = \langle \{x, y, z, \ldots\} : R^1, \ldots, R^n \rangle$$

The interpretation of the model is real-world entities and events and the aim is to account for real-world phenomena using the entities and relations in the model. So, some rules are needed for developing expressions derived from the model (syntax = formation rules) and ways are needed to connect the class of expressions from the model to real-world phenomena.

You can have models of any phenomena – the rotation of the moon, chemical reactions, numbers, ant behaviour,……

What are the limitations on SMLs

Requirements

What must our models be able to do?

Components of the Models

Minimally, we need a set of signs fulfilling specific functions. ***How can we do that? What is presupposed?***

We assume discrete and sequential units – why do we make those assumptions and what do they tell us?

Questions

Which simultaneous features are there in natural languages? Why are they of limited use?

Why are non-discrete features “marginal”?

The “linear” sequence is a temporal sequence. What constraints does the temporal sequence bring?

A simple case:

Suppose we want to represent what exists. We need:

A sign of existence – an actualiser – E

A set of names of identifiable parts of reality – {*river, house, fish*, ….} = {s}

= the pair of sets <E, {s}>

And the connection between the two – i.e. a combinatory relation R. i.e. the model

<E, {s}: R>

And a formation rule – combine E with any single s.

What do we get?

Could E be expressed in different ways?

What do we find in real languages? Why?

How can we extend the model?

How about Negation?

We need to introduce N = negation – but where? ***Give two possibilities***:

Differences?

How about Questions? What do we need and what is the effect?

By introducing questions, we introduce a need for a discourse rule – i.e. how to respond to a question.

And we need to amend the rules for combination and sequence

Replies might be: Es, EsN.

What happens in real languages for interrogation and reply? How would the different possibilities be represented in an extended model?

Why?

How do we include "yes/no" responses? What rules are needed?

Consider how the model would be extended to deal with simple location.

Or, judgements:

Negation ([N]) is optional ([]) but requires, as above, an additional ordering of the combinatory relations.

Making suggestions/giving instructions – devise a model.

e.g. I = *Let's*

What other possibilities are there in real languages?

<{a}, I; R>

{actions} = {*go, drink, swim, run, eat*,….}

<{I}, {a}; R>

How is I expressed?

Suppose

{a^1} = {*give, fetch, eat, drink*,….}

What else do we need?

Here we introduce the possibility of an optional s – *give food, drink water, give, drink*.

If we include *run, swim* etc. s may be obligatorily absent. Thus creating a different possibility in the model with a classification of the {actions}.

Notice:

Some models overlap. {s}, {actions}/{Instructions}

Some sets are limited to one SML – Suggestion, {Judgements}

Further Considerations

Limited distributions are diagnostic of the structure and are hence "nuclear" – *exist, let's*, judgement good.

Multiple distributions have variable functions and must be dependent on nuclear signs.

Some nuclear signs are limited to a few (or just one) signs (*Let's*); others are open inventories. Role of contrasting closed and open inventories as well as different inventories according to function.

Joining SMLs together is more efficient with overlaps – same signs re-cycled in different contexts. But it implies having function markers.

Is this always the case with natural languages? How do signs differ according to function/context? Why should this be so?

How would this model differ from the model for asserting existence/non-existence? How might it overlap?