

Re: Universal grammar

Source: <http://sci.tech--archive.net/Archive/sci.lang/2006-10/msg01552.html>

- *From:* Tak To <takto@xxxxxxxxxxxxxxxx>
 - *Date:* Mon, 23 Oct 2006 12:20:48 -0400
-

Hans Aberg wrote:

Tak To <takto@xxxxxxxxxxxxxxxx> wrote:

In any case, I think "a universal representation of meaning" exist, which is the collectively state of neurons in one's brain. It is just that it is not that convenient to "use".

I am playing around with the same ideas, but strictly in the domain of mathematics. Let me drop off some inputs:

Traditional metamathematics typically (i.e., most, but not all) treat object math theories as sequences of strings. But mathematicians do not agree on notation, even though they generally agree on the notions, i.e., the semantics.

One might attempt to pin down this semantics using tree structures similar to ASTs (abstract syntax tree – readers not knowing this stuff might use the Wikipedia). Take the expression $A \Rightarrow B$, which may be given prefix (Lukasiewicz) notation " $A B \Rightarrow$ " or RPN " $\Rightarrow A B$ "; but the AST is the same:

```
=>
/\
A B
```

Sorry, I am not following you at all.

I don't know what your "object math theories" are. (I know of a branch of computing science that tries to lay a mathematical foundation of "objects" as in "object oriented programming". Most of the work in that field is based on lambda calculus and has nothing in common with what you have described.)

If you are talking about mathematical logic that deals with "propositions", "theorems", "truth values", etc, then I don't

Re: Universal grammar

know of any problem related to notations.

Have you heard of Early's Algorithm,

Those tend to be slow. Do you know how GLR works?

My memory is vague. Isn't the worst case of GLR a lot worse than that of Earley's? In any case, I was just throwing alternative ideas.

Take the classical example:

Time flies like an arrow, fruit flies like a banana.

Both parts are ambiguous, here. It is not difficult to write a parser that keeps track of all possibilities. But this way, all the possible parses of the first part is s_1, \dots, s_k , and the second part is t_1, \dots, t_l , get combined the expansion of the possibilities $(s_1, \dots, s_k) \times (t_1, \dots, t_l)$, or $k \cdot l$ items.

By contrast, suppose the GLR grammar merges at the "," in the example above. Then one is forced to write an action, or semantics description, of the first s_1, \dots, s_k possibilities. The parser then starts over with one single parsing branch to get the t_1, \dots, t_l possibilities. When these merge, one gets a semantic description of $(s_1, \dots, s_k) \times (t_1, \dots, t_l)$, and the parsing is much quicker, as it does not have to carry all those branching possibilities along after the merges. Natural languages seem to have this kind of local ambiguities.

But this is hardly a worst case for GLR!?

Oh, well, this is just a side issue.

So I think this answers your questions. Yes, one is forced to develop logical models for the natural language semantics.

I am still not sure what you plan to do with the output of the parser. For example, do you plan to feed it into an inference engine so that you can answer questions about the input sentence?

Tak

Re: Universal grammar

Re: Universal grammar

Tak To takto@xxxxxxxxxxxxxxxx

[taode takto ~{LU5B~}] NB: trim the xx to get my real email addr