Re: Universal grammar

Source: http://sci.tech−archive.net/Archive/sci.lang/2006−10/msg01325.html

- *From*: haberg@xxxxxxxxxx (Hans Aberg)
- *Date*: Thu, 19 Oct 2006 06:22:12 GMT

In article <1161231426.615399.55320@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
groups@xxxxxxxxxxxxxxxxxxx wrote:

> > The original question was actually about linguistic grammar universals, not language universals. I looked up Joseph Greenberg, and he seems to have been into language classification, rather than finding a universal grammar theory. So I suspect, though interesting, it might not be useful.
>
> I wouldn't spend too much time looking for those grammatical universals either if I were you. The whole issue is a trip wire for doctrinal affiliations, but I think it is safe to say, at least, that there is not the consensus about it within linguistics which you would want before you tried to carry any supposed insights over into your theorem prover.

The theorem prover is a wholly separate project from the parsing ideas.
But ideas and principles from that might be transported to other contexts.

> What is funny about Universal Grammar is that arguably it was only proposed because linguistic universals were _not_ seen in natural language. They were not seen to the extent that observable entities did not even seem to be consistent throughout a single language (e.g. the way phonemes sometimes have one sound value and sometimes another.) If structure general to a language, the very building blocks of the language, could not be observed, the argument went, it cannot be learned, and if it cannot be learned it must be innate.
>
> The trouble is, it proved impossible to posit workable universals on theoretical grounds either. Optimality Theory in the '90s might be thought of as a last attempt to describe a level at which they might work. Wikipedia: 'The main idea of OT is that the observed, "surface" forms of the language arise from the resolution of conflicts between grammatical constraints.' But at that point they seem to have been reduced to a level of "universality" so deep, it would be hard to distinguish from the neural substrate. Indeed Wikipedia now tells me

> what I did not know, that OT owed its origins at least as much to connectionism as generative grammar.

Thank you for the additional references for me to look at.

I looked at:
http://en.wikipedia.org/wiki/Optimality_theory

The funny thing is that there is a parallel development in computer science in the 1990s, where operator predecenses can be expressed using grammar constraints. There I made a result where I generalized to a concept of a CFG grammar with such constraints, then showing it can be rewritten as a CFG. Thus, one does not get a larger formal grammar class. It is just a method of describing the grammar used, which is in itself of course important, giving structure and simplifying the writing of grammars.

> The search for the fundamental building blocks of natural language keeps grabbing handfuls of air.

The problem, I think is to get hold of the semantics, which is expressed via the language. It is the same in computer languages, and most metamathematics, thus taking a syntactic approach. But object (= working) mathematics, is structured around the semantics of common mathematical objects (like integers, real numbers, etc.).

> > I write on a theorem prover. And as pure mathematicians do no agree on notation, I build it up around certain semantic constructs, which the parser can translate into. It then does not matter exactly what the input language is, if only the parser is set right. And one can write out in different notation, if one has the opposite of the parser, called "expresser" perhaps.
>
> I can understand why you might think semantic constructs could provide you with the universals you need. That was the way one branch of generative grammar developed. Look up Generative Semantics. It collapsed under the wealth of detail in the '70s and largely became a theory of lexicon (Cognitive Grammar) or metaphor (Lakoff.)
>
> The search for universals has come full circle and it has become a canon of Cognitive Linguistics that to capture the wealth of semantic detail parametrizing language, only the actual language in use suffices.
>
> No−one has been able to find a universal representation for meaning, any more than they have been able to find other universals.

> But you are right. I think the issue is very closely related to that of a general theorem prover in maths. Actually, I didn't think it was still considered possible to build a general theorem prover in maths (and I am personally convinced that is for the same reason we have not been able to find universals of language or meaning.)

My practical scope is merely to write something that could help doing mathematicians doing the proof checking. This must then involve theorem proving, because doing a formal proof in all its axiomatic details quickly become unworkable to humans.

But I have been able to do automated induction proofs, and it does a breadth first proof tree search, meaning that all proofs are searched through. So if a proof exists, it can in principle be found. Ib reality, though memory and time will quickly run out for even simple statements. And if a proof does not exist, nontermination will result. So the proof will be a method of cutting the search possibilities, but that is all that is needed.

The advantage with a theorem prover, compared to the parsing of a human language, is that it is relatively easy to pin down semantics representations: just invoke some standard metamathematics (though I have moved beyond that).

But suppose one would want a parser parsing human (natural language) written mathematics. Then the same problems as when parsing other human languages will occur. There, a more realistic goal is to find a computer language making it comfortable for humans to write proofs. But this is still very far from being able to express typical pure mathematics.

> I have been recommending to linguists that they look at this talk by Greg Chaitin:
>
> http://www.cs.auckland.ac.nz/CDMTCS/chaitin/cmu.html
>
> Didn't Goedel prove back in the '30s that a general theorem prover was not possible?

He proved that in an axiomatic theory essentially containing the natural numbers as an entity, there are true statements that cannot be proved. If one would have such a statement P, and tries to prove P and not P in parallel, the theorem prover would end up in nontermination. So the theorem prover cannot decide whether it is such a statement. But one does not know of any such explicit statement in working math.

> Or have I mistaken what you mean by "theorem prover"?

In a theorem prover, one plugs in a statement, and it finds the proof. In a proof checker, if one has a finished proof, and it checks the correctness. For practical purposes, a human would need a hybrid, because (as I said before) a detailed formal proof is not practically workable to humans.

--
Hans Aberg
.