

Re: Universal grammar

Source: <http://sci.tech-archive.net/Archive/sci.lang/2006-10/msg01508.html>

- *From:* Joachim Pense <snob@xxxxxxxxxxxxxxxx>
 - *Date:* Sun, 22 Oct 2006 12:13:53 +0200
-

Am 20 Oct 2006 08:50:53 -0700 schrieb Rob Freeman:

Hans,

A long message. In the hopes of finding some consensus I've reduced it down to the bit we agree on. That makes the discussion much shorter :-)

Hans Aberg wrote:

In article <1161318846.624985.11180@xx>, "Rob Freeman" <groups@xx> wrote:

Is it possible to reduce all of mathematics to logic?

...

Is it possible to reduce all patterns to logic?

...

I don't know about you, but I'm starting to think "incompressible string".

I do not know what you mean by this term. – I start thinking about computer data compression. :-)

Good, good, Hans. We agree on something!! This is related to the compressibility of strings.

The question, rephrased again, is, are all strings compressible?

Re: Universal grammar

Every string can be compressed to a single bit. Just take this
Algorithm: Let S to be the string I want to compress to one bit. So if
I have any string T , just compare T with S . If T is equal to S , just
emit one "1" bit. If it is not equal, emit T as-is, but with a "0" bit
prepended.

Of course this algorithm is not particularly good :-), because it
compresses exactly one string to one bit, while every other string is
actually expanded by one bit.

The bottom line is: It doesn't make much sense to talk about the
compressibility of single strings (or finite sets of strings). What
you want to compress is always an infinite class of strings (English
books, or executable programs), and the payoff will always be that
there will be strings outside this class that will be expanded rather
than shrunk. (This can be very easily shown by a counting argument)

Of course, the actual theory is much smarter – you don't just identify
a class of strings – rather you produce a statistical model that is
constructed in a way that the type of strings you want to compress is
a highly probable output of your modeled generator, while the other
strings are not.

Joachim

.