Universal Grammar Is a Universal Grammar

by Ramón Casares

Universal Grammar is a universal grammar, because the human brain circuitry that implements the faculty of language is Turing complete.

Keywords: Universal Grammar, Turing completeness, language evolution

§1 Introduction

¶1 · I was aware for many years of the different use of the word 'universal' in computing theory, as used in 'universal Turing machine', and in linguistics, as used in 'Universal Grammar'. That was the reason why I was using the term 'syntax engine' for the brain circuitry that implements Turing completeness, instead of using any better expression with the word 'universal' in it. And it is also the reason why I have written this paper: *Could it be that both 'universals' are nevertheless the same?*

¶2 · From [Chomsky 1959] we deduce that for each grammar there is a Turing machine, and conversely. Following this equivalence, it is immediate to conclude that a universal Turing machine is equivalent to a universal grammar. This should settle the question, but, in linguistics, Universal Grammar is the human brain circuitry that implements the faculty of language, see [Chomsky 2005]. So the definitive resolution is achieved only when we show that the human brain is Turing complete, see [Turing 1936], and that language uses this capability. Therefore the answer is *yes*: Universal Grammar is a universal grammar.

§2 Hierarchy

¶1 · Chomsky presented, in [Chomsky 1959], a hierarchy of grammars. A *grammar* of a language is a device that is capable of enumerating all the language sentences. And, in this context, *language* is the (usually infinite) set of all the valid sentences.

¶2 · At the end of SECTION 2 in that paper, we read: "A type 0 grammar (language) is one that is unrestricted. Type 0 grammars are essentially Turing machines". At the beginning of SECTION 3, we find two theorems.

THEOREM 1. For both grammars and languages, type 0 $\supseteq$ type 1 $\supseteq$ type 2 $\supseteq$ type 3.

THEOREM 2. Every recursively enumerable set of strings is a type 0 language (and conversely).

This is draft version 20140411.
Any comments on it to papa@ramoncasares.com are welcome.

¶3 · Then THEOREM 2 is explained: "That is, a grammar of type 0 is a device with the generative power of a Turing machine."

¶4 · From the two theorems we can deduce three corollaries.

COROLLARY 1. The set of all type 0 grammars (languages) is equal to the set of all grammars (languages).
This is because, according to THEOREM 1, type 0 is the superset of all grammars (languages).

COROLLARY 2. For each Turing machine there is a type 0 grammar (and conversely).
This is equivalent to THEOREM 2, but in terms of grammars (devices) instead of languages (sets).

COROLLARY 3. For each Turing machine there is a grammar (and conversely).
This results by applying COROLLARY 1 to COROLLARY 2.

¶5 · Then, according to COROLLARY 3, for each Turing machine there is a grammar, and conversely, and therefore for each universal Turing machine, which can behave as any Turing machine, there is a universal grammar, which can behave as any grammar, and conversely. Summarizing, there is a universal grammar, which is equivalent to a universal Turing machine.

§3 Factors

¶1 · In computing theory, a universal grammar is a device that can behave as any grammar, and therefore it is capable of generating any possible language, while in linguistics, Universal Grammar is a theoretical concept posited to explain why humans acquire language, but other species do not. So we will write Universal Grammar, capitalized, to refer to the linguistic concept, and universal grammar, all in lower case, to refer to the computing theory device. A universal grammar is then a mathematical concept, so it is well-defined, while Universal Grammar is a concept in linguistics no so well defined. Then our next task will be to get a better definition of Universal Grammar.

¶2 · According to [Chomsky 2005], there are three factors that explain "the growth of language in the individual", that is, the acquisition of language:
- genetic endowment,
- experience, and
- other principles not specific to the faculty of language.

Universal Grammar is what results from the development of the first factor, genetic endowment, and this means that Universal Grammar is hardwired in the human body, mainly in the human brain. This explains why a human individual, exposed to enough linguistic experience, acquire language, and why a chimpanzee does not. The different linguistic experiences are responsible for the different natural languages.

¶3 · We are now ready to state its definition: *Universal Grammar* is the human brain circuitry that implements the faculty of language. And then, to characterize Universal Grammar, we have to investigate the linguistic capabilities of the human brain.

¶4 · But, before going on, please note that we have used the material assumption to define Universal Grammar. The *material assumption* can be stated like this: when a physical object performs a behavior that provides a function, we assume that the function is somehow instantiated physically in the object. We then say that the object, or a part of it, implements the function. You can consider that the material assumption describes what really happens, or just use the assumption as a linguistic device that allows you to refer metaphorically to a function as if it were a thing.

§4 Computing

¶1 · To assess the human brain capabilities we will examine our computing capabilities.

¶2 · When Turing wrote his paper, in 1936, a computer was a person. So the Turing machine, as it was presented in [Turing 1936], models the calculations done by a human computer. This means that we can compute whatever a Turing machine can compute, provided we have enough time and memory, and therefore we are Turing complete.

¶3 · This time we have explicitly expressed the proviso, but in what follows we will omit it, in the understanding that the proviso 'provided it has enough time and memory' is tacitly stated whenever we refer to a finite computing device.

¶4 · And now, using the material assumption again, from the fact that our brain is Turing complete, we assume that there is some circuitry inside our brain that implements Turing completeness.

¶5 · Invoking Church's thesis, see [Arbib 1987], being Turing complete is being able to compute any possible algorithm, or, in other words, a device is Turing complete if and only if it can follow any finite set of well-defined rules. So Turing completeness is the pan-computing capability, because it is the capability of executing any possible computation. We are Turing complete, so our brain capabilities are then the maximum computing capabilities.

§5 Completeness

¶1 · We have seen that our brain capabilities are the maximum computing capabilities, but here we are only interested in our linguistic capabilities. So we must consider the possibility that language does not use the full computing capabilities of the brain, and, more precisely, that the Universal Grammar does not use the brain circuitry that implements Turing completeness. We will call this possibility the *incomplete hypothesis*. Now I will show you that it is wrong, so the opposite possibility, which we will call the *complete hypothesis*, is right.

¶2 · If the incomplete hypothesis were right, then language would be unable to use some mathematical expressions. Some would say that this is the case, showing some artificial constructions not found in any natural language. But firstly note that, being able to show them, although in quotes, means that they can be used. And secondly see that artificial languages used in computing also use constructions not found in any natural language, and that, anyway, these artificial constructions can be internalized and used easily, effortless, and unconsciously, by good programmers.

¶3 · I do not think it would be fruitful to suggest that good programmers are not using their language faculty when they write programs in some computing language, because, in fact, those good programmers use naturally a lot of those artificial constructions when they talk to each other. Summarizing, natural languages can incorporate any artificial construction.

¶4 · The incomplete hypothesis has also serious difficulties with evolution. The supporters of the incomplete hypothesis should explain, without even mentioning language, why our pan-computing capability evolved, and they should also explain why language does not use that pan-computing capability, that is, Turing completeness, being available for free. Their task is very difficult because computing has a close relationship with language and grammar, as shown in [Chomsky 1959].

¶5 · In a similar way, it would also be very difficult for the supporters of the incomplete hypothesis to explain why two human capabilities, language and mathematics, which are both peculiarities only found in humans, have nevertheless evolved separately.

¶6 · For these reasons, I disregard the incomplete hypothesis. See that, in the end, this only means that I consider the Turing complete brain circuitry as part of the Universal Grammar.

§6 Heterogeneity

¶1 · Nevertheless, there are some other aspects not yet considered, as efficiency, or difficulty. It could be that we compute some algorithms more efficiently than other algorithms of the same computational complexity, because we can follow some kinds of rules more easily than other kinds. We will call this possibility the *heterogeneous hypothesis*. If that were the case, then natural languages would tend to evolve towards our easier rules.

¶2 · Please note that the opposite, which we will call the *homogeneous hypothesis*, is also possible. It could be that we find any rule as easy, or as difficult, as any other rule. The Turing machine is a very good example, because Turing machines use only one type of rule: if the current state is s, and the symbol read is i, then next state will be n, the written symbol will be o, and the next movement will be m. Therefore a universal Turing machine uses only one type of rule, and yet it is the prototype of Turing completeness. This just proves that homogeneous Turing completeness is possible.

¶3 · The homogeneous hypothesis draws a flat algorithmic landscape, where every algorithm is as easy, or as difficult, to execute as any other, while for the heterogeneous hypothesis the landscape is hilly, with difficult peaks and easy valleys. An evolution of algorithms, in a hilly landscape, would be seen as a trajectory that is being canalized to the lower lands, while in a flat landscape only the fitness of the algorithms would have any effect, and thus the trajectory would not be canalized.

¶4 · That not everybody can become a good programmer suggests that our species is still evolving to homogeneous Turing completeness. This is because it seems that for good programmers the homogeneous hypothesis is the case, while for everybody else the case is the heterogeneous hypothesis. Some caution is nevertheless needed because, if there is a critical period to acquire language, during which a child can learn any language, then it is also possible that within that period the homogeneous hypothesis stands for everybody.

¶5 · On the other hand, and for the same reasons that the brain automatize frequent behaviors, genetic evolution can take a beneficial behavior and code it into the genome. This could have happened to language acquisition after achieving Turing completeness.

Then we would find some restrictions that would help children to acquire their languages, while the faculty of language would still use the Turing complete circuitry. So this explanation would imply more canalization, that is, more heterogeneity, during the critical period.

¶6 · What these last two contradictory explanations show is that there is a trade-off between homogeneity and heterogeneity that evolution has to engineer. Most evolutionary trade-offs depend on the environment. In this case, if the environment favors programming, mathematics, and problem resolving (see [Casares 2014]), then evolution will select Turing complete homogeneity, while if the environment favors language acquisition and easiness, then evolution will instead select some heterogeneity.

§7 Conclusion

¶1 · Disregarding the incomplete hypothesis implies that our Turing complete brain circuitry is part of the brain circuitry that implements our faculty of language. Calling every Turing complete circuitry a *syntax engine*, we can abbreviate the implication: a syntax engine is part of the Universal Grammar. The conclusion is then that the Universal Grammar, using its syntax engine, can behave as any grammar. In other words, *Universal Grammar is a universal grammar*.

¶2 · This conclusion, Universal Grammar is a universal grammar, means that natural language grammars are not limited, although they are probably canalized towards easier rules and towards easier acquisition, and each one is surely conditioned by its own history. This happens because Universal Grammar is evolving as a compromise that takes into account its various functions and its implementation. Universal Grammar functions include, at least, language, mathematics, and problem resolving, and its implementation includes its interaction with other brain modules, and other engineering conditions.

¶3 · That we are still evolving, and that evolution is not finished but still running, should not surprise anybody. But, at least in this respect, the situation should now be clearer, considering Turing completeness as a fixed point in the evolution of language.

References

[Arbib 1987] Michael Arbib, *Brains, Machines, and Mathematics*, Second Edition; Springer-Verlag, New York, 1987, ISBN: 978-0-387-96539-0.

[Casares 2014] Ramón Casares, "Resolution Machinery"; `http://cogprints.org/9209/`.

[Chomsky 1959] Noam Chomsky, "On Certain Formal Properties of Grammars"; in *Information and Control*, Volume 2, Issue 2, pp. 137–167, June 1959.

[Chomsky 2005] Noam Chomsky, "Three Factors in Language Design"; in *Linguistic Inquiry*, Volume 36, Number 1, pp. 1–22, Winter 2005.

[Turing 1936] Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem"; in *Proceedings of the London Mathematical Society*, Volume s2-42, Issue 1, pp. 230–265, 1937. Received 28 May, 1936. Read 12 November, 1936.

Longer abstract

Is Universal Grammar a universal grammar? From Chomsky's hierarchy we deduce that for each grammar there is a Turing machine, and conversely. Following this equivalence, it is immediate to conclude that a universal Turing machine is equivalent to a universal grammar. Meanwhile, in linguistics, Universal Grammar is the human brain circuitry that implements the faculty of language. So the definitive answer is achieved only when we show that the human brain is Turing complete, and that language uses this capability. So yes: Universal Grammar is a universal grammar, because the human brain circuitry that implements the faculty of language is Turing complete. (102 words)