Evelyn Drake

Dr. Demeter

ENGL 301

27 March 2025

Constructed Language Project – 16Lang

16Lang is my constructed language, and it was born from my passion for computer science. The name comes from the hexadecimal number system, which is primarily used in computing. Instead of the base 10 digits `0-9`, the base 16 digits go from `0-9` and then `A-F` for a total of 16 values per character. My language uses this system to convey information far more concisely than English without losing much meaning or specificity. This is especially important in the context of artificial intelligence, which is becoming increasingly adopted in nearly every discipline. I am not particularly thrilled about artificial intelligence, especially because of its significant environmental impact. By representing concepts more concisely, 16Lang will be able to alleviate a variety of issues with large language models. Because it can be mapped to most languages nearly one-to-one, a model can have its training data translated before the process starts, allowing it to use data from a larger variety of sources from different languages. Similarly, a model's output can be easily translated into any desired language. Large language models work by breaking phrases into "tokens," which are smaller subunits of a fixed size that are then considered and predicted. In most computational representations of English, each letter is assigned a 2-digit hexadecimal number, and then words are built up from these characters (e.g. "Hi" becomes `48 69`). By mapping hexadecimal numbers directly to concepts instead of representing each English character with a unique number, the storage overhead required to represent ideas shrinks dramatically.

While I will describe the writing system in great deal in the remainder of the paper, here is an example to prove its efficiency: Using ASCII character encoding, the hexadecimal equivalent of the phrase “moved quickly” is 6D 6F 76 65 64 20 71 75 69 63 6B 6C 79, which requires 104 binary bits (ones and zeroes). In 16Lang, the equivalent of this phrase is <1x20> 2x10. Assuming we use 2 bits (which can store 4 values, although there are only 3 tenses) to indicate the verb tenses and represent the number before the x as 4 bits (16 possible values), this phrase requires only 26 bits—4 (first number of first word) + 8 (second number of first word) + 2 (verb tense) + 4 (first number of second word) + 8 (second number of second word)—drastically decreasing the length of the stored information by a factor of 4!

The writing system of this language is logographic, and the only valid characters are the digits 0-F. It cannot be pronounced phonetically as it is not intended to replace oral communication. All words are constructed from two numbers separated by the character x, (number 1 x number 2). The first number must be one digit, which represents the word’s part of speech. The second number can be an arbitrarily long series of digits, which represent increasingly specific concepts as more digits are added. For nouns, which begin with 0x followed by a number, the initial second digits are 1: Matter and Substances, 2: Living Beings, 3: Time and Space, 4: Abstract Concepts, 5: Social Structures, 6: Objects and Tools, 7: Natural Phenomena, 8: Motion and Energy, 9: Numbers and Quantities, A: Senses and Perception, B: Emotions, C: Body Parts and Functions, D: Knowledge and Learning, E: Spirituality, and F: Miscellaneous. I have included a table to demonstrate how this works:

	0	1	2	3	4
0x1	Earth	Air	Water	Metal	…

0x2	Human	Animal	Plant	Bacteria	…
0x3	Moment	Second	Year	Location	…
0x4	Truth	Idea	Love	Hatred	…
0x5	Family	Government	Law	Economy	…
0x6	House	Book	Computer	Table	…
…					

As previously mentioned, the concepts become increasingly specific as more numbers are added; for example, adding numbers onto the end of 0x21, the concept of animals, leads to the following words: 0x210: Cat, 0x211: Dog, 0x212: Cow, etc. More suffixes can be added endlessly to expand the language's lexicon, with the idea being that the most common words would be given shorter numbers in an ideal implementation of 16Lang.

For verbs, which begin with 1x followed by a number, the initial second digits are 1: Existence, 2: Motion, 3: Perception, 4: Thought and Speech, 5: Emotion, 6: Physical Manipulation, 7: Social Interaction, 8: Change, 9: Measurement and Judgement, and A: Necessity. Again, I have included a table to demonstrate how this works:

	0	1	2	3
1x1	To be	To become	To seem	…
1x2	To move	To stop	To run	…
1x3	To see	To hear	To feel	…
1x4	To think	To say	To ask	…
1x5	To love	To hate	To enjoy	…
1x6	To take	To give	To carry	…

…				

For adjectives, which begin with 2x followed by a number, the initial second digits are 1: Physical Properties, 2: Sensory Qualities, 3: Emotional States, 4: Moral Qualities, 5: Utility, and 6: Comparisons. Consider the following table:

	0	1	2	3
2x1	Big	Small	Fast	…
2x2	Loud	Bright	Cold	…
2x3	Happy	Sad	Fearful	…
2x4	Good	Evil	Just	…
2x5	Useful	Broken	Useless	…
2x6	Similar	Different	Equal	…
…				

For adverbs, which begin with 3x followed by a number, the initial second digits are 1: Manner, 2: Frequency, 3: Degree, 4: Time, and 5: Location. Once again, I have included a table for reference:

	0	1	2	3
2x1	Quickly	Slowly	Carefully	…
2x2	Often	Rarely	Always	…
2x3	Completely	Somewhat	Barely	…
2x4	Now	Later	Soon	…

2x5	Here	There	Everywhere	…
…				

This pattern continues for the remaining parts of speech. I will not include tables for every part, as it should now be obvious how the words are constructed from the two numbers. The last categories are 4x: Determiners (the, this, that, some, all, etc.), 5x: Degree words (very, more, less, enough, etc.), 6x: Auxiliary verbs (can, must, will, might, etc.), 7x: Prepositions (in, on, under, beside, etc.), 8x: Conjunctions (and, or, but, because, etc.), and 9x: Pronouns (I/me, you, he/him/his, she/her/hers, they/them/theirs, etc.). Additionally, plurality is indicated at the end of the noun following a colon, which is represented by a numerical quantity in hexadecimal: part of speech x word: number. The number 0 is reserved for the equivalent of "-s" in English (if the word is singular, no colon follows it). Punctuation simply follows each word when necessary (.,?!"). The character ~ can be added to the end of a word to denote possession (the equivalent of "-'s" in English). Tense can be indicated by wrapping verbs in angle brackets (<>: past tense), curly brackets ({}: present tense), and square brackets ([]: future tense). The present tense is inferred when a verb is not enclosed in brackets. To use the phrase "to move" (1x20) as an example, <1x20> translates to "moved," {1x20} translates to "moves/is moving," and [1x20] translates to "will move."

This is an exclusively analytic language, as it relies entirely on word order instead of spoken inflections to indicate grammatical relationships. The word order is similar to English's and follows the subject-verb-object pattern. Adjectives precede nouns, adverbs can come before or after verbs, determiners precede nouns, prepositions precede their objects, and conjunctions connect clauses and phrases. Here are some example sentences and phrases: 4x0 0x210

<1x22> 2x10. translates to "The cat ran quickly." 4x0 0x201 {1x50} 4x0 0x200. translates to "The woman loves the man." 2x50 {1x1} 0x212:0 2x52! translates to "There are cows everywhere!" 0x90 2x10 {1x20} 0x61:68. translates to "I quickly moved 104 books." Note that in this example, the hexadecimal number 68 following the colon is equivalent to 104 in decimal, as each place value holds 16 digits instead of 10 (16*6+1*8=104). 9x2 6x1 {1x32} 0x43. translates to "She must feel hatred."

Overall, designing this language has been both challenging and exciting. The most interesting part was designing a numbering system that could represent both linguistic structure and semantic meaning efficiently, and it seemed intuitive enough to me that I was surprised that a similar constructed language didn't already exist (to my knowledge). The most difficult part of this process was ensuring that words progress in specificity as more digits are added. Of course, this language is far too impractical for humans to communicate with, and it cannot be verbally pronounced. The idea behind it is that given a sufficiently large word bank that is mapped onto different numbers, English and other similar languages can be easily translated to 16Lang due to its familiar syntax, which can then be used in computing applications (like large language models) to gain massive reductions in storage and processing complexity. Because of its analytical grammar and its ability to endlessly expand in meaning numerically, it is very easy to create compound words and derive new words from existing words. This project has been an interesting experience for me, and I loved having the opportunity to combine my passion for computer science with my growing interest in linguistics.