Representative Artificials On Letter Combinatorics Case With Predicate Sentences.

Frank Appiah

November 13, 2020

Representative Artificials On Letter Combinatorics Case With Predicate Sentences.

Prof Frank Appiah AKA FAEng PhD
appiahnsiahfrank@gmail.com
8.10.2020

Abstract. This review is an investigation reference on letter combinatorics showing the table information of s-index, predicate sentences and more as artificial representations in a five enumerated sentence case example.

Keywords. s-index, sentence, problem solving, words, Tableaux, combinatorics.

1 INTRODUCTION

Letter combinatorics is about sentences or phrases and counting problems. It is logical structured and involves discrete operations like subtraction, addition and multiplication. It is about alphanumeric labeling of sentences or phrases and proofing of combinatorial enumerations. The theory of combinatorics of sentences or phrases or words is called Letter Combinatorics (LC) with 8 bulletin requirements. A Marriage Problem (MP) made up of 5 sentences is used in the exploit of letter combinatorics. A generating function is calculated for MP to handle constraints of arrangement /selection and the combinatorial enumerations of MP are also calculated. This example of letter combinatorics shows the calculation of addition , multiplication and subtraction principles of MP.

Letter Combinatorics has or must have the following requirements:

- A countable number of sentences or phrases.

- Counting the size of selected phrases or likely occurrence of subset equality of letters is called count.
- A sentence with the number of letters specified is called Π.
- The logical structure of arithmetic such as +, and = should be applied.
- Discrete operations must include count, addition, subtraction and sizes.
- The sizes of selected phrases are enumerated.
- Proofs with the discrete operations on which the enumeration of sizes stops.
- There is a possibility of summation equal to Π.

2 MARRIAGE PROBLEM (MP EXAMPLE)

(1) Damn it.
(2) What's wrong?
(3) It is a combination of 46 letters.
(4) Akua will not marry you.
(5) Pokua will not marry you.

I will illustrate the size graphics and model a size representation for each sentence.

MP-1 Artificial

Illustration 1: Sentence (1) Size Graphics

MP-2 Artificial

Illustration 2: Sentence (2) Size Graphics

MP-3 Artificial

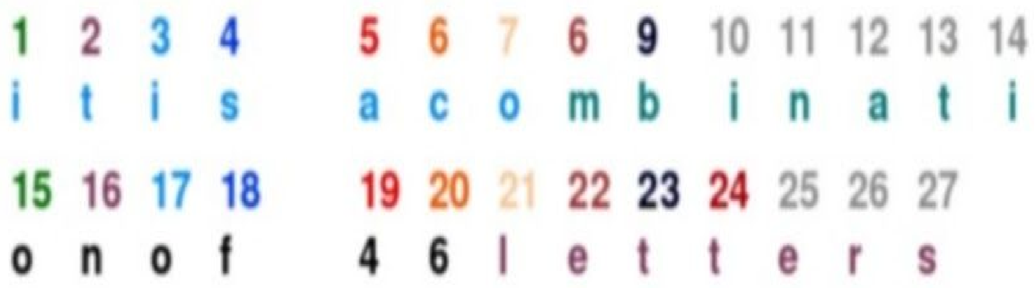

Illustration 3: Sentence (3) Size Graphics

MP-4 Artificial

Illustration 4: Sentence (4) Size Graphics

MP-5 Artificial

Illustration 5: Sentence (5) Size Graphics

The size graphics is a representation of artificial models alike an Evolutionary computation approach. This concept is based on list /array processing ideas. Representation artificials are colored versions of evolutionary computation. A s-index value starting from 1 to the length of sentence letters excluding white spaces are incrementally considered. These

artificial models of the sentences are color representation of all the 5 sentences. Each s-index is placed on top of each alphabet or letter without full stop consideration.
The minimum s-index is 1 and maximum is length of sentence strings. Natural language processing is the sort of artificial intell-igence being designed in this AI review work.

S-Index Computation

Sentence Number	Minimum Letters	Maximum Letters	Word Length
1	1	6	2
2	1	10	2
3	1	27	7
4	1	19	5
5	1	20	5

3 PREDICATE SENTENCES

The MP sentences are represented as predicates with each word captured in the predicate sentence, *mpsentence*. The following are the predicate sentences for the MP example :

1. mpsentence(damn, it).
2. *mpsentence(what's, wrong).*
3. *mpsentence(it, is, a, combination, of, 46, letters).*
4. *mpsentence(akua, will, not, marry, you).*
5. *mpsentence(pokua, will, not, marry, you).*

General predicate : **mpsentence (word_1, word_n), n > 1.**

The next predicate is to determine if a sentence is a question or not. There is only one question in all the five sentences. It is represented as *mpsentenceask* predicate sentence. This will take on two passing values of sentence number and an indicator of a question or not. The following question stances are:

1. *mpsentenceask(1,no).*
2. *mpsentenceask(2, yes).*
3. *mpsentenceask(3, no).*
4. *mpsentenceask(4, no).*
5. *mpsentenceask(5, no).*

General Predicate : **mpsentenceask (sentence _no, response).**

The number of words of a sentence is now represented with *mpwordsize* predicate sentences. . The following details are as follows :

1. *mpwordsize(1, 2).*
2. *mpwordsize(2, 2).*
3. *mpwordsize(3, 6).*
4. *mpwordsize(4, 5).*
5. *mpwordsize(5, 5).*

This predicate took its arguments to be the sentence number and the number of words. General predicate is represented as:

General Predicate : **mpwordsize (sentence_no, word_number).**

Further details on negation sentences are looked at. This will have the predicate sentence, *mpnegation*. This is explicitly sentences with a **not** word.
The problem solution are as follows :

1. *mpnegation(1, no).*
2. *mpnegation(2, no).*
3. *mpnegation(3,no).*
4. *mpnegation(4, yes).*
5. *mpnegation(5, yes).*

General Predicate : **mpnegation (sentence _no, response).**

MP example has only two negation statements in total. Statements like "damn it" creates a feeling of regret or disappointment. What's wrong did create sudden worry but does not bring the negation that is not interesting. The predicate sentence is represented as *mpregret*. These are as follows :

1. mpregret(1, yes).
2. *mpregret(2, no).*
3. *mpregret(3, no).*
4. *mpregret(4, no).*
5. *mpregret(5, no).*

Reference Work : Turing AI World Leading Review. 2020

General Predicate : **mpregret (sentence _no, response).**

mpworry is the predicate sentence for sudden worry. These includes the following :

- *mpworry(1, no).*
- *mpworry(2, yes).*
- *mpworry(3, no).*
- *mpworry(4, no).*
- *mpworry(5, no).*

General Predicate : **mpworry (sentence _no, response).**

The problem solver took on statement 3 to bring out an approach. The predicate for this will be *mpsolver*. The knowledge needed to be programmed are as follows:

1. *mpsolver(1, no).*
2. *mpsolver(2, no).*
3. *mpsolver(3, yes).*
4. *mpsolver(4, no).*
5. *mpsolver(5, no).*

General Predicate : **mpsolver (sentence _no, response).**

The third round tried to bring out a solution in the context of problem solving. The 4 and 5 statements are involved with names of female sex. These are Akua and Pokua. The fact base for this representation is captured with predicate sentences, *mpnamsex*. Theses will include the following :

- *mpnamsex(1, no).*
- *mpnamsex(2, no).*
- *mpnamsex(3, no).*
- *mpnamsex(4, yes).*
- *mpnamsex(5, yes).*

General Predicate : **mpnamsex (sentence _no, response).**

It will be smart to know of the exact names involved. *mpname* predicate will be used to store facts of name information. These includes the following sentences:

1. *mpname(1, people).*
2. *mpname(2, object).*
3. *mpname(3, thing).*
4. *mpname(4, person).*
5. *mpname(5, person).*

General Predicate : **mpname (sentence _no, response).**

This predicate captures a person's fact to the database. The assertions are as follows :

- *mpperson(1, noname).*
- *mpperson(2, noname).*
- *mpperson(3, noname).*
- *mpperson(4. Akua).*
- *mpperson(2, Pokua).*

General Predicate : **mpperson (sentence _no, response).**

The name information brings out the predicate concepts that includes *mpstate* that combines the words people, person, object and thing to the sentences.

The following statements are made:

- *mpstate(1, 'Damn it on people').*
- *mpstate(2, 'What's wrong with you').*
- *mpstate(3, 'The thing is a combination of 46 letters').*
- *mpstate(4, 'A person will not marry you').*
- *mpstate(5, 'A person will not marry you').*

General Predicate : **mpstate(sentence _no, response).**

4 Conclusion.

The Joy of predicates on 5 Secondary sentences is done in conclusion remarks.
Finally, the s-index predicate sentences are enumerated below :

1. sindex(1, 1, 6, 2).
2. sindex(2, 1, 10, 2).
3. sindex(3, 1, 27, 7).
4. sindex (4, 1, 19, 5).
5. sindex(5, 1, 20, 5).

General Predicate : **sindex (sentence _no, min_letter, max_letter, word_count).**

Reference Work : Turing AI World Leading Review. 2020

Further Reading.

(1) Appiah F. "Letter Combinatorics : Theory on counting problems ". EPSRC UK Turing AI Letter. 2020
(2) Appiah F. "Letter Combinatorics : Theory on counting problems ". Marriage Problem. Mendeley Publication. 2020
(3) Hooper, Joan B. "On assertive predicates." In Syntax and Semantics volume 4, pp. 91-124. Brill, 1975.
(4) Yoon, Youngeun. "Total and partial predicates and the weak and strong interpretations." Natural language semantics 4, no. 3 (1996): 217-236.
(5) Shoben, Edward J., Keith T. Wescourt, and Edward E. Smith. "Sentence verification, sentence recognition, and the semantic-episodic distinction." Journal of Experimental Psychology: Human Learning and Memory 4, no. 4 (1978): 304.
(6) Dechaine, R. A. "Predicates across categories: Towards a category-neutral syntax." (1994): 0501-0501.
(7) Aarts, Bas. "5 Secondary predicates in English1." The verb in contemporary English: Theory and description (2006): 75.
(8) Sagot, Benoît. "Linguistic facts as predicates over ranges of the sentence." In International Conference on Logical Aspects of Computational Linguistics, pp. 271-286. Springer, Berlin, Heidelberg, 2005.
(9) Kratzer, Angelika. "Individual-level predicates." The generic book 125 (1995).
(10) Graff, Delia. "Descriptions as predicates." Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition 102, no. 1 (2001): 1-42.
(11) Huang, C-T. James. "Complex predicates in control." In Control and grammar, pp. 109-147. Springer, Dordrecht, 1992.
(12) Farkas, Donka. "Two types of world-creating predicates." The Joy of Grammar, John Benjamins (1992): 35-7.
(13) Peperkamp, Sharon, and Emmanuel Dupoux. "Learning the mapping from surface to underlying representations in an artificial language." Laboratory phonology 9 (2007): 315-338.
(14) Johansson, Tobias. "Strengthening the case for stimulus-specificity in artificial grammar learning: no evidence for abstract representations with extended exposure." Experimental Psychology 56, no. 3 (2009): 188-197.