A Base Number Representation on Marriage Problem Predicate Task.

Frank Appiah

April 8, 2021

Report Reference

A Base Number Representation on Marriage Problem Predicate Task.

Prof Frank Appiah AKA FAEng PhD
appiahnsiahfrank@gmail.com
5.04.2021

Abstract. This report is an investigation reference on letter combinatorics showing the predicate sentences in base number representation and setting a table representation of a binary set. In here, octal representation in base - 8, decimal representation in base-10 and binary representation in base-2 are calculated for the binary equivalents of the 1/0 set values.

Keywords. sentence, base, representation, words, table, set, binary number.

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. The predicate sentences are made from [5]. This work looks at binary number concepts on representation of predicates.
This research is organised as follows :

(1) A Look again on the predicate sentences with binary number representation ,
(2) Generate a Tableaux representation from the set[6,7, 8] form,
(3) Apply arithmetics operation on the set state and
(4) Make base representations of the set state

The Marriage Problem states that;

(1) Damn it.
Binary Representation : **01000100 01100001 01101101 01101110 00100000 01101001 01110100 00101110.**

(2) What's wrong?
Binary Representation : **01010111 01101000 01100001 01110100 00100111 01110011 00100000 01110111 01110010 01101111 01101110 01100111 00111111**.

(3) It is a combination of 46 letters.
Binary Representation : **01001001 01110100 00100000 01101001 01110011 00100000 01100001 00100000 01100011 01101111 01101101 01100010 01101001 01101110 01100001 01110100 01101001 01101111 01101110 00100000 01101111 01100110 00100000 00110100 00110110 00100000 01101100 01100101 01110100 01110100 01100101 01110010 01110011 00101110.**

(4) Akua will not marry you.
Binary Representation : **01000001 01101011 01110101 01100001 00100000 01110111 01101001 01101100 01101100 00100000 01101110 01101111 01110100 00100000 01101101 01100001 01110010 01110010 01111001 00100000 01111001 01101111 01110101 00101110 00100000.**

(5) Pokua will not marry you.
Binary Representation : **01010000 01101111 01101011 01110101 01100001 00100000 01110111 01101001 01101100 01101100 00100000 01101110 01101111 01110100 00100000 01101101 01100001 01110010 01110010 01111001 00100000 01111001 01101111 01110101 00101110 00100000.**

2 Binary Number Representation

The MP sentences are represented as predicates with each word captured in the predicate sentence, mpsentence(MpS). The following predicate sentences for the MP example are in [5]. This category predicate is important in this work.
These category predicates will be represented as follows in set forms:

```
1. mpsentence(damn, it).
MpS1={damn, it}.
MbS1={01000100 01100001 01101101 01101110 00100000, 01001001
01110100 }.
```

2. mpsentence(what's, wrong).
MpS2={what's, wrong}.
MbS2={01010111 01101000 01100001 01110100 00100111 01110011, 01010111 01110010 01101111 01101110 01100111}.

3. mpsentence(it, is, a, combination, of, 46, letters).
MpS3={it, is, a, combination, of, 46, letters}.
MbS4={01001001 01110100, 01101001 01110011, 01100001, 01000011 01101111 01101101 01100010 01101001 01101110 01100001 01110100 01101001 01101111 01101110, 01001111 01100110, 00110100 00110110, 01101100 01100101 01110100 01110100 01100101 01110010 01110011}.

4. mpsentence(akua, will, not, marry, you).
MpS4={akua, will, not, marry, you}.
MbS4={01000001 01101011 01110101 01100001, 01110111 01101001 01101100 01101100, 01101110 01101111 01110100, 01101101 01100001 01110010 01110010 01111001, 01111001 01101111 01110101}.

5. mpsentence(pokua, will, not, marry, you).
MpS5={pokua, will, not, marry, you}.
MbS5={01110000 01101111 01101011 01110101 01100001, 01110111 01101001 01101100 01101100, 01101110 01101111 01110100, 01101101 01100001 01110010 01110010 01111001, 01111001 01101111 01110101}.

MpS={MpS1, MpS2, MpS3, MpS4, MpS5}

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 category predicate is important in this work.
This will take on two passing values of sentence number and an indicator of a question or not. Yes(Y will be 1) indicates a pass value whiles No(N will be 0) does 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).

In generating a set for mpsentenceask(named as MpA) , It will give:
MpA={N, Y, N, N, N}.
MbA={0, 1, 0, 0, 0}.

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 category predicate is important in this work. The set theoretic form is represented as :`
`MpWs={1.2, 2.2, 3.6, 4.5, 5.5}.`
`MbWs={1, 1, 1, 1, 1}.`

The set values are changed to decimal forms to indicate index of values. This so because sets does not accept the same values on indexing.

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).`

`The set representation of Mpnegation is`
`MpNg={N, N, N, Y,  Y}.`
`MbNg={0, 0, 0, 1, 1}.`

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).
```

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

`The set theoretical form is given by:`
`MpR={Y,  N,  N,  N,  N}.`

MbR={1, 0, 0, 0, 0}

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 set theoretical form is given by:
MpW={N, Y, N, N, N}.
MbW={0, 1, 0, 0, 0}.

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 set theoretical form is given by:
MpS={N, N, Y, N, N}.
MbS={0, 0, 1, 0, 0}.

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. These 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).

The set theoretical form is given by:
MpX={N, N, N, Y, Y}.
MbX={0, 0, 0, 1, 1}.

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).

The set theoretical form is given by:
MpR={P,  O,  T,  E,  E},  where p is people, o is object, t is
thing and e person.
MbR={01010000, 01001111, 01010100, 01000101, 01000101}.
```

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 set theoretical form is given by:
MpP={N.1,  N.2,  N.3,  A,  P}.
MbP={01001110 00101110 00110001, 01001110 00101110 00110010,
01001110 00101110 00110011, 01000001, 01010000}.
```

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).

The set results is represented by:
```

MpT={'1.Damn it on people', '2.What's wrong with you', '3.The thing is a combination of 46 letters', '4.A person will not marry you', '5.A person will not marry you'}

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

Si1={1, 6, 2}
Sb1={**0001**, 0110, 0010}

Si2={1, 10, 2}
Sb1={**0001**, 0110, 0010}

Si3={1, 27, 7}
Sb1={00010, **11011** , **00111**}

Si4={1, 19, 5}
Sb1={**00001**, **10011**, 0**0101**}

Si5={1, 20, 5}
Sb1={**00001**, **10100**, 0**0010**}

The following set operations are calculated on Si sets:
(1) Unions: Si1 U Si2 U Si3 U Si4 U Si5= {1, 2, 5, 6, 7, 10, 19, 20}
(2) Si1 intersect Si2= {1, 2}
(3) Si2 intersect Si3= {1}
(4) Si3 intersect Si4= {1}
(4) Si4 intersect Si5= {1, 5}
(5) Si1 intersect Si2 intersect Si3 intersect Si4 intersect Si5= {1}.

Si={Si1, Si2, Si3, Si4, Si5}

The following are used in forming Tableaux representation :
MpA={N, Y, N, N, N},
MpNg={N, N, N, Y, Y},
MpR={Y, N, N, N, N},
MpW={N, Y, N, N, N},
MpX={N, N, N, Y, Y},

MpS={N, N, Y, N, N},

Y/N Tableaux Representations

No	MpA	MpNg	MpR	MpW	MpX	MpS
1	N	N	Y	N	N	N
2	Y	N	N	Y	N	N
3	N	N	N	N	N	Y
4	N	Y	N	N	Y	N
5	N	Y	N	N	Y	N

The following are used in forming Tableaux representation :
MbA={0, 1, 0, 0, 0}.
MbX={0, 0, 0, 1, 1}.
MbS={0, 0, 1, 0, 0}.
MbR={1, 0, 0, 0, 0}
MbW={0, 1, 0, 0, 0}.
MbNg={0, 0, 0, 1, 1}.

1/0 Binary Representations

No	MbA	MbNg	MbR	MbW	MbX	MbS
N1	0	0	1	0	0	0
N2	1	0	0	1	0	0
N3	0	0	0	0	0	1
N4	0	1	0	0	1	0
N5	0	1	0	0	1	0

3 Base Operation

The arithmetics field of algebra will be performed on the six sets which will transformed into values in assessment.
The arithmetic operations are as follows :

- A + B
- A - B
- A x B
- A / B
- A % B

The set will be equalised to the following in base 2:

- MbA=01000
- MbX=00011
- MbS=00100
- MbR=10000
- MbW=01000
- MbNg=00011

The representation in base 10 [9] are as follows :

- MbA=8
- MbX=3
- MbS=4
- MbR=16
- MbW=8
- MbNg=3

The arithmetic operation will be done in base -10 and then in base- 2 to base - 8.

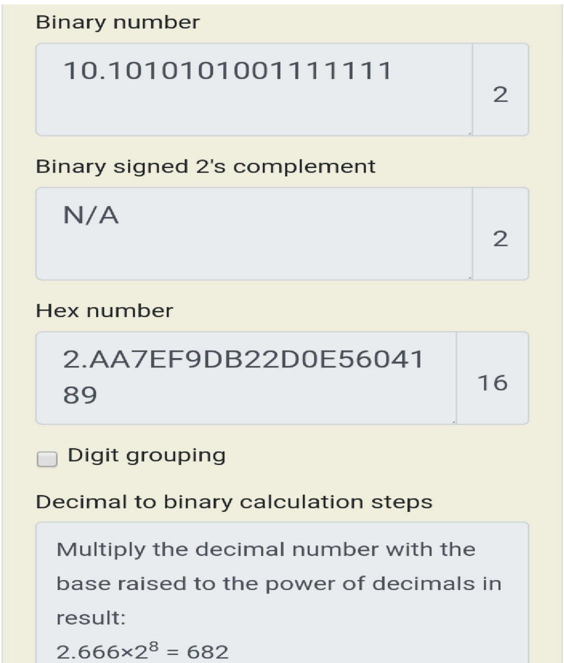
Base-10 Representation

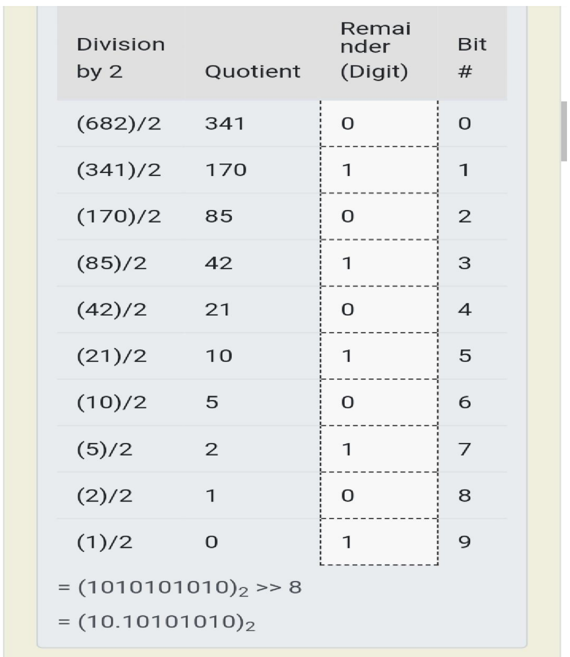
Operation	MbA operand	MbX operand	Result
-	8	3	5
+	8	3	11
x	8	3	24
/	8	3	2.6666
%	8	3	5

Result in base Representations

Base-10	Base-2	Base-8
5	00101	5
11	01011	13
24	11000	30
2.666	10.10101010	2.52477371
5	00101	5

Calculating 2.666 to binary

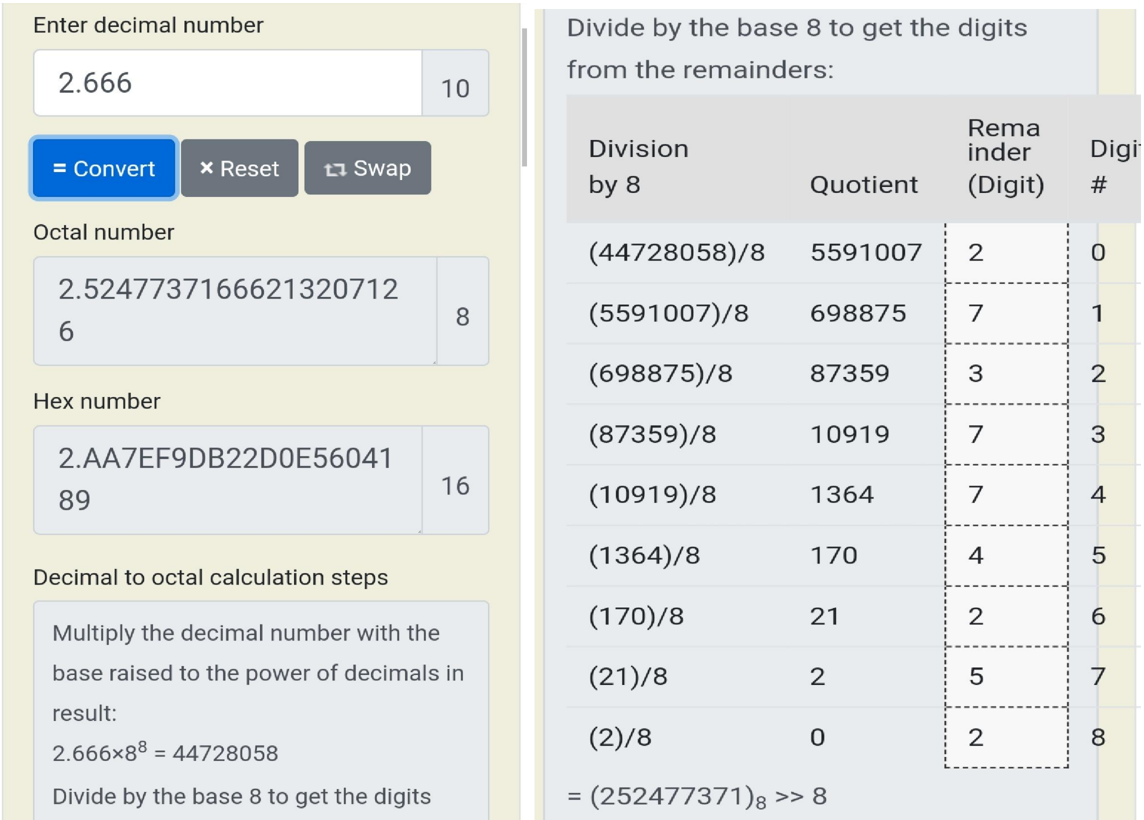




Division by 2	Quotient	Remainder (Digit)	Bit #
(682)/2	341	0	0
(341)/2	170	1	1
(170)/2	85	0	2
(85)/2	42	1	3
(42)/2	21	0	4
(21)/2	10	1	5
(10)/2	5	0	6
(5)/2	2	1	7
(2)/2	1	0	8
(1)/2	0	1	9

$= (1010101010)_2 >> 8$
$= (10.10101010)_2$

Converting 2.666 to octal representation



Representation Table

Name	Base 2	Base 10	Base 8
MbA	01000	8	10
MbX	00011	3	3
MbS	00100	4	4
MbR	10000	16	20
MbW	01000	8	10
MbNg	00011	3	3

The calculations for the base-8 in table are shown below :

Octal number

3 | 8

Hex number

3 | 16

Decimal to octal calculation steps

Divide by the base 8 to get the digits from the remainders:

Division by 8	Quotient	Remainder (Digit)	Digit #
(3)/8	0	3	0

= $(3)_8$

Octal number

10 | 8

Hex number

8 | 16

Decimal to octal calculation steps

Divide by the base 8 to get the digits from the remainders:

Division by 8	Quotient	Remainder (Digit)	Digit #
(8)/8	1	0	0
(1)/8	0	1	1

= $(10)_8$

Octal number

4 | 8

Hex number

4 | 16

Decimal to octal calculation steps

Divide by the base 8 to get the digits from the remainders:

Division by 8	Quotient	Remainder (Digit)	Digit #
(4)/8	0	4	0

= $(4)_8$

Octal number

20 | 8

Hex number

10 | 16

Decimal to octal calculation steps

Divide by the base 8 to get the digits from the remainders:

Division by 8	Quotient	Remainder (Digit)	Digit #
(16)/8	2	0	0
(2)/8	0	2	1

= $(20)_8$

4 Conclusion

This work on binary representation and Computer arithmetics concludes with the following remarks:

- Six 1/0 response set are achieved.
- Five non-response set are achieved.
- Table representation of the 1/0 set is achieved.
- Binary operation on s-index is achieved.
- S-index set has 5 subset in achieving.
- The MpS set has 5 member subsets.
- MpWs set is a binary number memberset.
- Base representation in 10, 8 and 2 are achieved.

Further Reading.

(1) Appiah Frank. Letter Combinatorics : Theory on counting problems. EPSRC UK Turing AI Letter. 2020
(2) Appiah Frank. 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) Frank Appiah. Representative Artificials On LetterCombinatorics Case With Predicate Sentences. Easychair No 4556. November 13, 2020.
(6) Jech, T. (2013). Set theory. Springer Science & Business Media.
(7) Fraenkel, A. A., Bar-Hillel, Y., & Levy, A. (1973). Foundations of set theory. Elsevier.
(8) Enderton, H. B. (1977). Elements of set theory. Academic press.
(9) Number Unit Conversion. 2021. Accessed Online: http://www.unitconversion.org/numbers/binary-to-base-10-conversion.html
(10) String Conversion. 2021. Accessed Online : https://www.convertbinary.com/text-to-binary/