# Digital Logic on Marriage Problem Predicate Task.

Frank Appiah



April 14, 2021

# Digital Logic on Marriage Problem Predicate Task.

(Version 2)

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

**Abstract**. This report is about digital logic study of binary sets on Marriage Problem Predicate with its tabular representation defined over [1,0] sets, **B** with set of parameters from the sentence predicates of the case examples used in the research.

**Keywords**. sentence, digital logic , predicate name, tabular representation, binary set.

## 1 Introduction

Marriage Problem 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 set concepts, tabular representation of predicates in creating a binary set and applying digital Logic on the elements of set.

*The Marriage Problem states that;*

*(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.*

This research is organised as follows :

- Show the extended digital set of the predicate names,
- A Look again on the predicate sentences with Digital set on tabular representation.
- Perform a boolean operation on the digital values.

# 2 Binary Set on Tabular Representation

In answering queries whether a particular predicate is a Digital-1 or not. 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. 1 indicates a pass value whiles 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 MbA) ,  It will
give:
```

**MbA={0, 1, 0, 0, 0}.**

MbA is a binary set. 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 :
```

**MbWs={1, 1, 1, 1, 1}.**

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

**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:

**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:

**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:

**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:

**MbX={0, 0, 0, 1, 1}.**

The following are used in forming binary set on tabular 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}.

**Tabular Representations On Binary Set**

| B | 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 |

The parameters in the table of binary set with any pair forms a Digital set.

*For example Digital set (MbA, MbNg) will show the following :*

*Digital set (MbA, MbNg)={(0,0), (1,0), (0,0), (0,1), (0,1)}.*

*The binary logic will now be treated on the above Digital set.*

| MbA | MbNg | And | Or | Xor |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |

**The Digital logic((MbA, MbNg)=(and,or,xor)) ={(0,0)=(0,0,0), (1,0)=(0,1,1), (0,1)=(0,1,1)}**

*For example Digital set (MbA, MbR) will show the following :*

*set (MbA, MbR)={(0,1), (0,0), (0,0), (0,0), (0,0)}.*

*The binary logic will now be operator treated on the above Digital set.*

| **MbA** | **MbR** | **And** | **Or** | **Xor** |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**The Digital logic((MbA, MbR)=(and,or,xor)) ={(0,1)=(0,1,1), (1,0)=(0,1,1), (0,0)=(0,0,0)}**

*For example Digital set (MbA, MbW) will show the following :*

*Digital set (MbA, MbW)={(0,0), (0,1), (0,0), (0,0), (0,0)}.*

*The binary logic will now be operator treated on the above Digital set.*

| MbA | MbW | And | Or | Xor |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**The Digital logic((MbA, MbR)=(and,or,xor)) ={(0,0)=(0,0,0), (1,1)=(1,1,0)}**

*For example Digital set (MbA, MbX) will show the following :*

*Digital set (MbA, MbX)={(0,0), (0,0), (0,0), (0,1), (0,1)}.*

*The binary logic will now be operator treated on the above Digital set.*

| **MbA** | **MbX** | **And** | **Or** | **Xor** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |

**The Digital logic((MbA, MbR)=(and,or,xor)) ={(0,0)=(0,0,0), (1,0)=(0,1,1),(0,1)=(0,1,1)}**

*For example Digital set (MbA, MbS) will show the following :*

*Digital set (MbA, MbS)={(0,0), (0,0), (0,1), (0,0), (0,0)}.*

*The binary logic will now be operator treated on the above Digital set.*

| MbA | MbS | And | Or | Xor |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**The Digital logic((MbA, MbS)=(and,or,xor)) ={(0,0)=(0,0,0), (1,0)=(0,1,1),(0,1)=(0,1,1)}**

For example Digital set (MbNg, MbR) will show the following :

Digital set ( MbNg, MbR)={(0,1), (0,0), (0,0), (1,0), (1,0)}.

| MbNg | MbR | And | Or | Xor |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 |

**Digital logic((MbNg, MbR)=(and,or,xor)={(0,1)=(0,1,1), (1,0)=(0,1,1)}**

For example Digital set ( MbNg, MbW) will show the following :

Digital set ( MbNg, MbW)={(0,0), (0,1), (0,0), (1,0), (1,0)}.

| MbNg | MbW | And | Or | Xor |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |

**Digital logic((MbNg, MbW)=(and,or,xor)={(0,1)=(0,1,1), (1,0)=(0,1,1)}**

For example Digital set ( MbNg, MbX) will show the following :

Digital set ( MbNg, MbW)={(0,0), (0,0), (0,0), (1,1), (1,1)}.

| MbNg | MbX | And | Or | Xor |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |

**Digital logic((MbNg, MbX)=(and,or,xor)={(1,1)=(1,1,0)}**

For example Digital set ( MbNg, MbS) will show the following :

Digital set ( MbNg, MbS)={(0,0), (0,0), (0,1), (1,0), (1,0)}.

| MbNg | MbS | And | Or | Xor |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |

**Digital logic((MbNg, MbX)=(and,or,xor)={(0,1)=(0,1,1), (1,0)=(0,1,1)}**

For example Digital set ( MbR, MbW) will show the following :

Digital set ( MbR, MbW)={(1,0), (0,1), (0,0), (0,0), (0,0)}.

| MbR | MbW | And | Or | Xor |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**Digital logic((MbNg, MbX)=(and,or,xor)={(1,0)=(0,1,1), (0,1)=(0,1,1)}**

For example Digital set ( MbR, MbS) will show the following :

Digital set ( MbR, MbS)={(1,0), (0,0), (0,1), (0,0), (0,0)}.

| MbR | MbS | And | Or | Xor |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**Digital logic((MbR, MbS)=(and,or,xor)={(1,0)=(0,1,1), (0,1)=(0,1,1)}**

# 4 Conclusion

This work on Digital set and Digital logic concludes with the following remarks:

- Six Digital sets are achieved.
- Tabular representation of the Digital set is achieved.
- Six Digital logic 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.
(10) Stephen Brown and Zvonko Vranesic(2005). Fundamental of Digital Logic with Vhdl Design. 2 Edition.