Static and Dynamic Diagrams for Syllogistic

Sumanta Sarathi Sharma[1][0000-0003-1904-3492] and Varun Kumar Paliwal[2][0000-0002-2294-944X]

[1] School of Philosophy & Culture
Shri Mata Vaishno Devi University Katra, India
ss.sharma@smvdu.ac.in
[2] Department of Electrical & Electronics Engineering
Birla Institute of Technology & Science Pilani, India
2020ht01034@wilp.bits-pilani.ac.in

Abstract. The Revised Method of Minimal Representation (RMMR) is an alternate diagrammatic technique, which uses a static and a dynamic interface to test the validity/invalidity of a categorical syllogism. In this paper, we develop algorithms to depict a moving environment, which helps us effectively understand the state of affairs and line of reasoning with respect to syllogistic.

Keywords: Aristotelian Logic · Diagrammatic Logic · Dynamic Diagrams · MMR · Modern Logic · RMMR · Static Diagrams.

Introduction

Aristotle was familiar with the usage of diagrams for academic discourses[12]. Studies suggest that diagrams play a crucial role in human understanding and help people reason rapidly & accurately, easily, and effectively[1],[8],[15]. In the last decade, there has been a surge of works about logic diagrams[9],[10],[11] and diagrams for syllogistic[2],[4],[14],[16],[17],[18]. Also, we have seen some works on syllogisms and algorithms in the recent past[5],[7],[13]. The Method of Minimal Representation (MMR) (earlier conceived as the 'method of least representation') is a diagrammatic technique to test the validity of syllogisms[20]. MMR tests the validity of syllogisms in both Aristotelian and Modern logic formats, which is its unique characteristic[19]. Moreover, MMR plays a significant role in distinguishing perfect and imperfect syllogisms[21].

This paper introduces the revised method of minimal representation (RMMR), which uses a static and a dynamic environment for syllogistic problem-solving. The revision aims to show the line of reasoning and visually cognize all possible state-of-affairs between classes depicted by the subject and predicate terms of categorical propositions and eventually, in a syllogism.This paper has three sections. In the first section, we explicate the essential preliminaries of RMMR, including singly and multiply represented diagrams for propositions and discuss its static and dynamic interfaces here. We understand the working of RMMR by taking few examples in the second section. In the final section, we develop algorithms for RMMR.

1 Preliminaries of RMMR

In RMMR, a 'rectangle' with 'U' engraved inside represents the diagram site. We know that a syllogism has three terms, namely S (minor), P (major), and M (middle). For this, we use three colors, i.e., red, green, and blue for S, P, and M, respectively. In addition to this, a 'four arrowed pointer' (FAP) for showing the movement of a set denoted by S, P, or M, is also used.

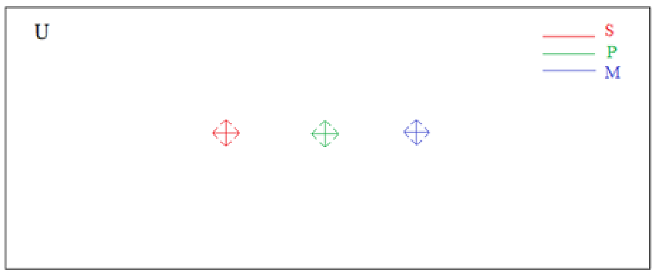

Fig. 1. Diagram site and FAP.

The 'Game of Logic' inspires FAP, which has pointers that move as per prescribed rules in the biliteral and triliteral diagrams to validate or invalidate the claimed conclusion[3]. In the same vein, the significance of FAP (in RMMR) is to show and restrict the movement of a set denoted by the corresponding color in a designated area. Also, the pointer tells that the FAP can move in any direction. In RMMR, we show a set (say S, P, or M) using a square with a diagonal inside. This set can be drawn anywhere inside the diagram site U.

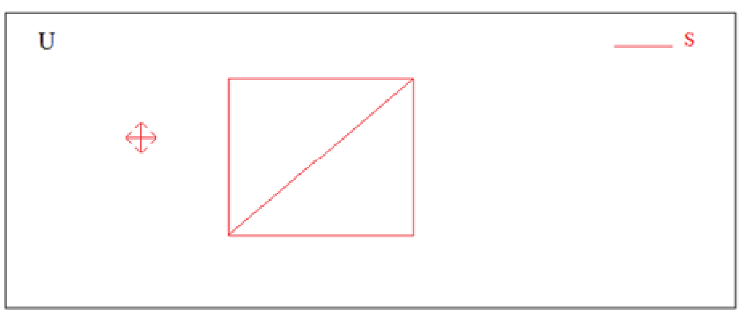

Fig. 2. Singly represented RMMR for any set.

The above diagram is a singly represented static diagram for various kind of movement, which is associated with the set S. A multiply represented static diagram of set S is depicted next:

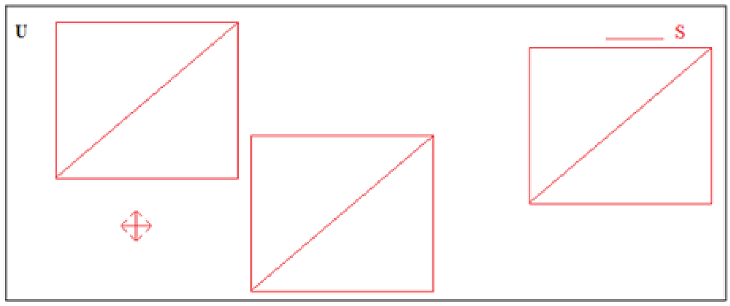

Fig. 3. Multiply represented RMMR for any set.

Both the above figures depict the possibility of set S in the diagram site U. The former is a singly represented static diagram. Thus, S is shown at one place. However, Fig. 3 is a multiply represented static diagram and thus, S has multiple representations of set S in the diagram site. Next, we draw propositions in RMMR.

1.1 Propositions in RMMR

There are four categorical propositions, namely universal affirmative, universal negative, particular affirmative, and particular negative, whose depictions are as under:

Universal Affirmative Proposition (A) 'all S is P' is represented as follows:

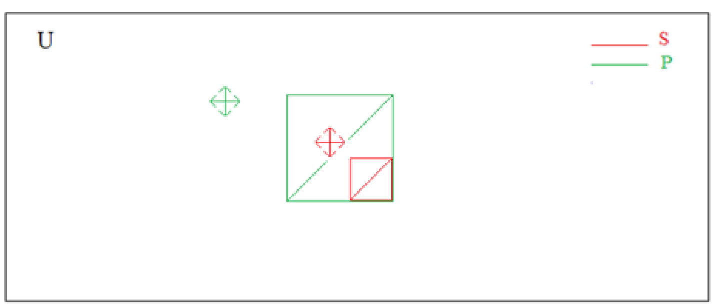

Fig. 4. RMMR for Universal Affirmative Proposition

Here, a smaller red square S is drawn inside a larger green square P to depict 'all S is P'. Moreover, red FAP S is placed inside the green square of P to show that the red square S can be present or move inside the green square only.

Moreover, it also tells us that the red square S cannot be present outside the green square P. However, green FAP P is placed outside the green square to show its movement and presence anywhere in the diagram site U. This diagram can have multiple iterations as S and P can be shown to have different sizes and can also acquire several positions in U. However, all the iterations must follow the rule as prescribed by FAP. As a matter of convention, we draw the smaller red square from the right side.

The position of FAP may be of some interest to the ratiocinator. It may be noted that we place the FAP inside or outside a corresponding set rather than the set itself. For example, a red FAP is placed corresponding to a green square and a green FAP corresponding to a red square. Since there are two sets in a proposition, one FAP each for red and green suffice our requirement. However, when we have to unify two propositions in a single diagram, we may require more than one FAP for a square (or a right-angled triangle, as the case may be).

Universal Negative Proposition (E) ‘no S is P’ is represented as follows:

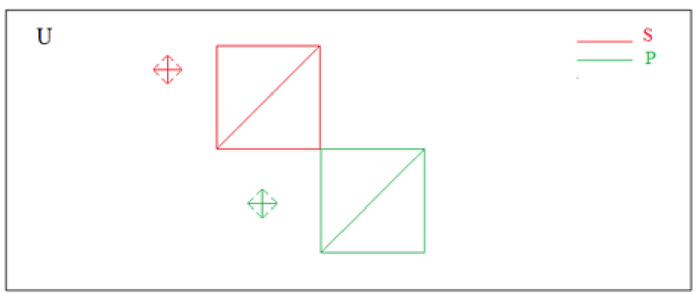

Fig. 5. RMMR for Universal Negative Proposition

Here, two non-overlapping squares, i.e., red S and green P, are drawn as shown. The green and red FAPs are also placed outside each other and inside U to show their respective positions. Thus, it shows that squares S and P can move away from each other but can neither intersect nor contain each other. This diagram too, can have multiple iterations as S and P can be shown to have different sizes and can acquire several positions in U as per the rules of FAP. In other words, squares S and P can never overlap or intersect with each other.

Particular Affirmative Proposition (I) To depict, ‘some S is P’, we draw a red right-angled triangle S inside the green square P. The rationale for choosing a right-angled triangle is to depict it as a bigger square. We have deliberately inserted the diagonal inside square to appear this incomplete square in a right-angled triangle. It is also a singly represented static diagram and can have multiple iterations. The red and green FAPs show that the red square S can only be inside P, whereas P can be anywhere in the diagram site U respectively.

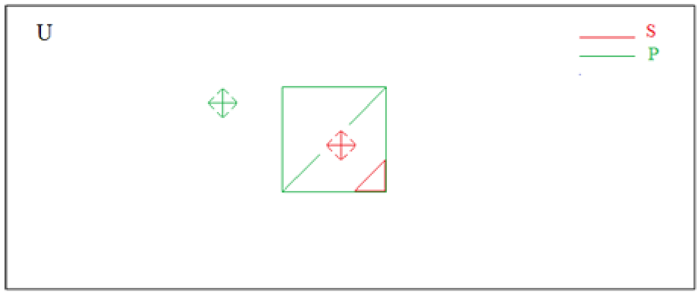

Fig. 6. RMMR for Particular Affirmative Proposition

Particular Negative Proposition (O) To diagram 'some S is not P', we draw a red right-angled triangle outside green square P. The red and green FAPs are placed outside green square and red right-angled triangle respectively to show that they cannot overlap in U.

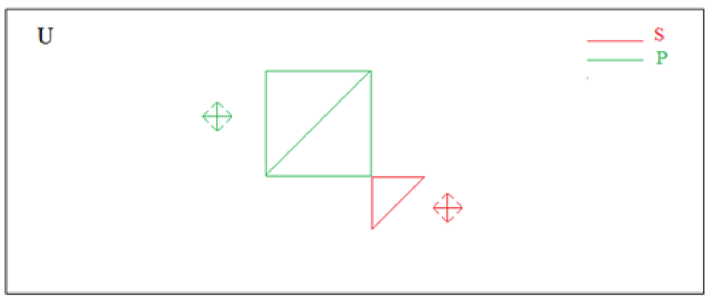

Fig. 7. RMMR for Particular Negative Proposition

All propositions in RMMR have multiple representations. This is akin to the concept of 'counterpart equivalence'[6], where two diagrams (*D* and *D'*) represent the same set of information and state of affairs except that it may have different size or orientation in the diagram site, they are called counterpart equivalent. All multiply represented static diagrams of a proposition depict counterpart equivalences of their singly represented static diagram. In other words, a multiply represented static diagram combines several diagrams in one diagram to express the equivalent state of affairs of its corresponding singly represented static diagram. The above expression demands some clarification from an inquisitive reader with respect to a singly represented static diagram, multiply represented static diagram, and counterpart equivalences. Let us understand it with the help of an example. Consider a universal negative proposition **E**. Some of its iterations are as shown:

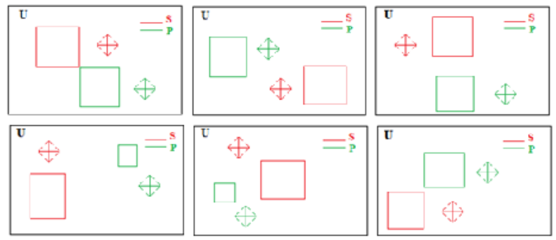

Fig. 8. Multiply represented RMMR for Universal Negative Proposition

These iterations, as shown above, are some of the depictions of a universal negative proposition in RMMR. Each of these iterations is a counterpart equivalent of the standard depiction of a universal negative proposition. The iterations also depict that the squares can be horizontally or vertically placed, have different alignments or sizes, and are not strictly squared. These iterations have to ensure that S and P are non-overlapping or non-intersecting closed four-sided figures. On the other hand, a multiply represented static diagram of a universal negative proposition is a combined diagram of these possibilities. In a way, it is like several screenshots of a moving environment plotted in diagram site U as shown below:

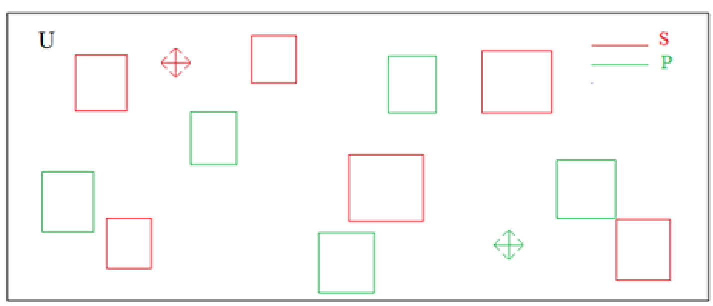

Fig. 9. Screesnshot of RMMR for Universal Negative Proposition

A multiply represented static diagram sketches some of the possibilities of finding a set in the diagram site. Roughly, it traces the movement of a set inside the diagram site. In the above case, sets S and P are shown to be found anywhere in U. However; they cannot intersect or overlap as depicted by FAPs. A multiply represented static diagram provides essential information to a ratiocinator while validating a syllogism. Let us see the working of RMMR to test syllogisms in the next section.

1.2 Testing syllogisms in RMMR

In this section, we will test the validity of syllogisms using RMMR. Before testing syllogisms, we explicate some general rules to draw the premises and read off the conclusion in RMMR.

Rules to draw premises in RMMR – A standard set of rules[22] to draw premises and validate a syllogism is as under:

1. Draw a single diagram to represent the facts that the two premises of a syllogism convey. (Let us call this diagram D_P.)
2. Draw a diagram to represent the fact that the conclusion of the syllogism conveys. (Let us call this diagram D_C.)
3. Check, if we can *read off* diagram D_C from diagram D_P.
4. If we can *read off* diagram D_C from diagram D_P then the given syllogism is valid or else invalid.

It is plain from above that *reading off* D_C from D_P holds the key in any diagrammatic technique for argumentation. D_P always contains more information than D_C since D_P combines two premises, whereas D_C depicts a single proposition. To *read off* D_C from D_P, we discount the surplus information of D_P and concentrate on the delineation of D_C. In what follows, we clearly state how to *read off* D_C from D_P.

Rules to *read off* conclusion in RMMR – In categorical syllogisms, we will have two propositions to be drawn on the diagram site as premises. It means that a combined diagram of two propositions has to be drawn. Let this process of combining the two diagrams be guided by the rules of unification. After successful unification, the next step would be to *read off* D_C from D_P. This will require us to subtract or delete excess information from D_P. Let us call them the rules of erasure. In what follows, we explain the laws of unification and omission.

Rule of Unification: Let D_1 and D_2 represent the major premise and minor premise respectively of a syllogism. Then D_P is called as the unification of D_1 and D_2 provided that D_P is the counterpart equivalence of D_1 and D_2 taken together. In other words, D_P is the sum total of all the information present in D_1 and D_2.

Rule of Omission: The diagram site contains S(red), P(green), and M(blue) parts. Erase the M (blue) part, either a rectangle or a right-angled triangle, from the diagram site U.

In what follows, we explain the working of the RMMR with some examples.

2 Working of the RMMR

In this section, we understand the working of RMMR with the static and dynamic diagrams. We will also see how dynamic diagrams play an indomitable and crucial role in testing syllogisms, which otherwise in a static diagram is hard and difficult to *read off*.

Let us begin with a simple example. Consider **AA** in the first figure, 'all M is P and all S is M', and then their corresponding D_1 and D_2 are shown as under:

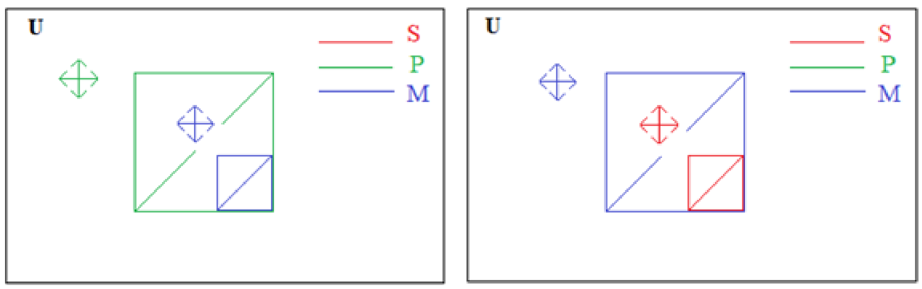

Fig. 10. AA in first figure

D_1 and D_2 can be unified together by placing the red square S inside the blue square M in D_1. The D_P so formed will have the FAPs in their respective positions, i.e., red inside blue, blue inside the green, and the green FAP inside U. The unified diagram is shown as under:

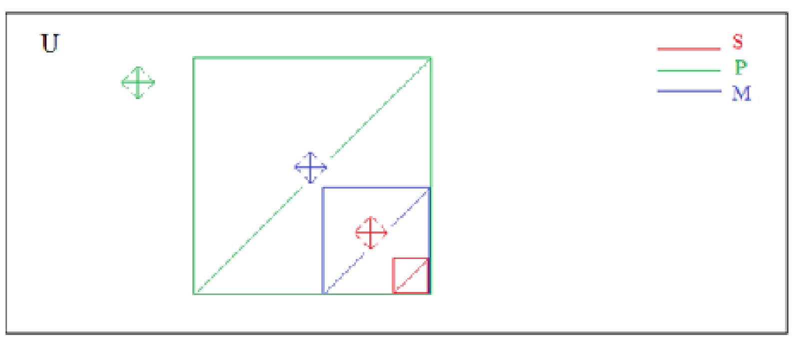

Fig. 11. Singly represented D_P of AA in first figure

Next, we apply the rule of omission in the above diagram by erasing the blue part, representing M. The diagram so formed is a counterpart equivalent of D_C, where we find a red square inside a green square (for all S is P) as shown in the following diagram.

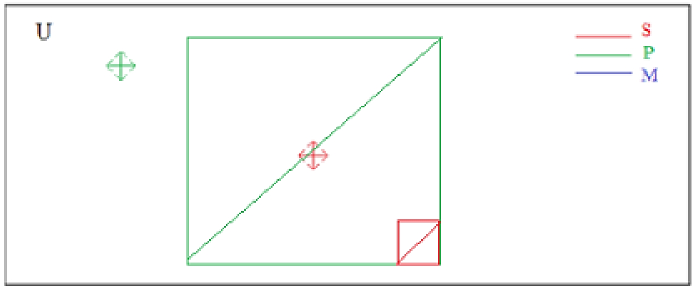

Fig. 12. Singly represented D_C of AAA in first figure

From the above diagram, we easily conclude, 'all S is P' from 'all M is P' and 'all S is M'. Now, let us consider a case where it is difficult to *read off* conclusion after drawing the premises. For instance, consider **AI** in the second figure where the premises are 'all P is M and some S is M'. Their corresponding D_1 and D_2 are shown as under:

Fig. 13. AI in first figure

D_1 and D_2 can be combined as follows:

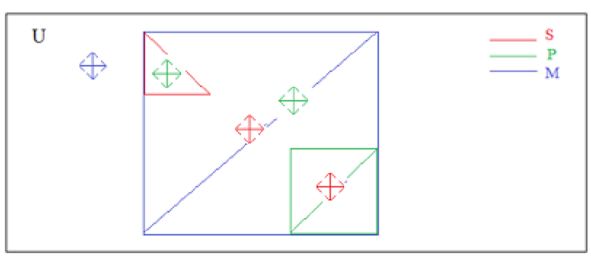

Fig. 14. Singly represented D_P of AI in second figure

We apply the rule of omission and erase the blue rectangle. The following is the obtained diagram:

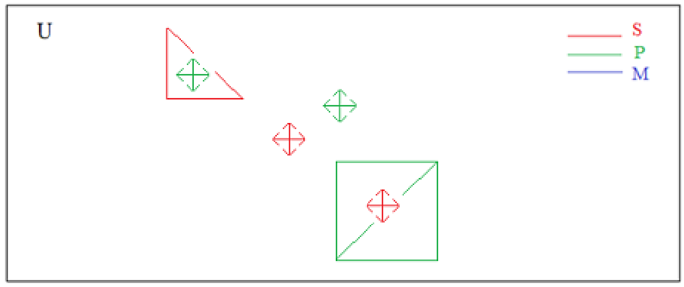

Fig. 15. Singly represented D_C of AI in second figure

An interpreter of the above diagram is tempted to conclude 'some S is not P'. However, the red and green FAPs show that the green rectangle P and the red right-angled triangle S can intersect and contain each other. Some possible states of the affair of their intersection and containment are given below:

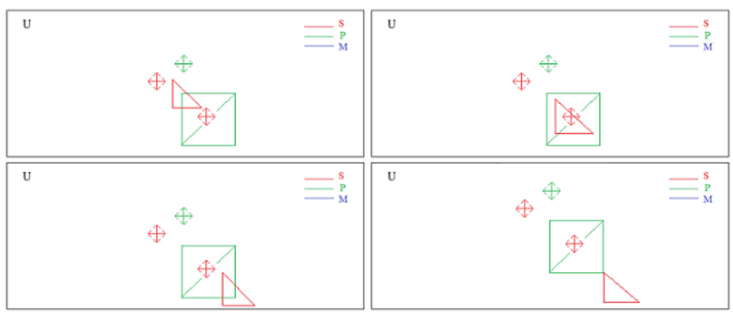

Fig. 16. Multiply represented D_C of AI in second figure

From the above diagram, it is clear that we cannot conclude that 'some S is not P'. We may reiterate that FAPs play an essential role in exhibiting the intersection or containment of the classes represented by the S, P, and M terms. As a matter of fact, we cannot conclude anything whatsoever from the above diagram in terms of categorical propositions, and thus, the premise pair of AI in the second figure does not yield any valid conclusion.

The authors admit that the above explanation or the expressed ideas (say, singly represented and multiply represented diagrams) seems sketchy and cumbersome (at least at the first go) as it is not clear to the ratiocinator that how to interpret or *read off* the conclusion from the D_P or D_C. It is an unavoidable situation initially, and to address this, we have devised a dynamic diagrammatic interpretation of the static diagrams. Therefore, let us reassert the static diagrams (both singly and multiply) and introduce the dynamic diagrams to an interpreter.

Static Diagrams – There are two types of static diagrams, viz. singly and multiply. Singly represented static diagrams (SRTD) are those diagrams, which represent a single (possible) state of affairs of a categorical proposition. Multiply represented static diagrams (MRSD) represent more than one (possible) state of affairs of a categorical proposition. However, it may be noted, that no MRSD can represent all possible states of affair, as it is infinite due to the size, shape, and orientation of rectangles, right-angled triangles, and the diagram site.

Dynamic Diagrams – Let the diagram site U is a moving environment. The squares and right-angled triangles can move inside the diagram site U as per the rules prescribed by FAPs. It generates dynamic diagrams of categorical propositions. In other words, a dynamic diagram is the total of all possibilities of relations expressed by the subject and predicate classes of categorical propositions. Another way of comprehending it will be to consider an MRSD as a finite number of screenshots (or say, graphical plotting) of the corresponding dynamic diagram.

It is impossible to plot a dynamic diagram on a sheet of paper. However, MRSD can always help us visualize the states of the affair between the subject and predicate classes. For example, an MRSD of EE in the fourth figure, i.e., 'no P is M' and 'no M is S' is represented as:

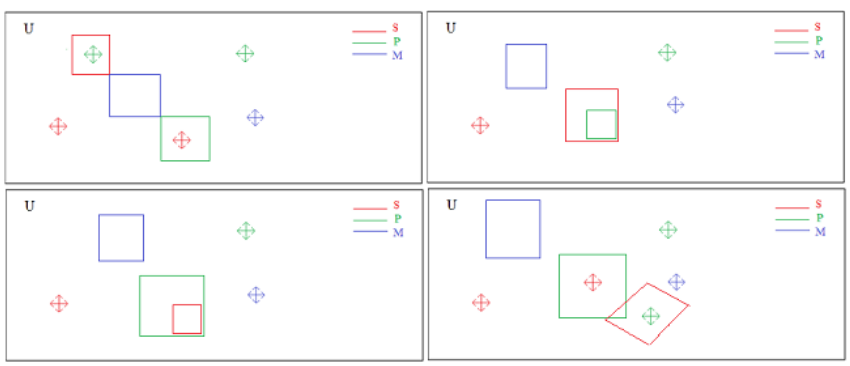

Fig. 17. Multiply represented D_P of EE in fourth figure

It can be seen from the above set of diagrams that S and P can intersect, contain as well as stay disjoint. Therefore, no conclusion follows from EE as premises in the fourth figure. The dynamic diagram of EE as premises is found here: RMMR for EE It is now clear (after clicking on the link) that there is an infinite number of ways in which MRSD can be depicted, and a dynamic diagram cannot be represented on a sheet of paper. SRSD, MSRD, and Dynamic diagrams may become intricate to draw diagrams for beginner and intermediate level reasoners. Thus, we introduce algorithms for the user to help her reason well.

3 Algorithms for RMMR

To help users draw the dynamic diagrams with ease, we pronounce the following rules for RMMR in the form of algorithms.

Algorithm 1: General RMMR

```
Data: Get proposition 1 (Major Premise) and proposition 2 (Minor Premise)
      and the proposition 3 (Conclusion) given.
Result: Decide whether the syllogism is "Valid" or "Invalid".
RMMR()
    Create a diagram site == U
    Obtain premise 1 and premise 2
    if (premise_2 == "A" || premise_2 == "E" ) && (premise_1 == "I" ||
     premise_1 == "O" ) then
        Diagram_Interpretation(premise_2)
        Another_Diagram_Interpretation (premise_1)
    else
        Diagram_Interpretation(premise_1)
        Another_Diagram_Interpretation (premise_2)
    end
    Validity_of_Conclusion()
end
```

The first step is to create the diagram site U. In the second step; we obtain the premises. After this, we check which of these premises is a universal proposition. If the first premise is universal, we draw it in the diagram site U. Or else, we draw the second premise (if it is a universal proposition). If both are universal, we draw any one of them. However, if both are particular propositions, we can again draw any one of the propositions first. The idea is to give preference to the universal proposition. The penultimate step concerns the interpretation of the premises, and finally, a conclusion is reached.

Algorithm 2: Diagram_Interpretation

```
Data: Get proposition which is universal affirmative or negative else take
      proposition 1.
Result: Dynamic diagrams for proposition.
Diagram_Interpretation(premise_type)
    if premise_type == "A" then
        Create square with diagonal which always reside in diagram site
        Create another square with diagonal which always reside inside the
         Initial square
```

```
    else if premise_type == "E" then
        Create square with diagonal which always reside in diagram site
        Create another square with diagonal which always reside outside
          the Initial square and always reside in diagram site
    else if premise_type == "I" then
        Create square with diagonal which always reside in diagram site
        Create right angled triangle which always reside inside the Initial
          square
    else if premise_type == "O" then
        Create square with diagonal which always reside in diagram site
        Create right angled triangle which always reside outside the
          Initial square and always reside in diagram site
```

In what follows, we first write the algorithm for drawing the second premise after the first premise. Some cases may require creating a square from a right-angled triangle in case of particular propositions. Next, we write the algorithm for validating the conclusion from the premises.

Algorithm 3: Another_Diagram_Interpretation

```
Data: Get another proposition.
Result: Dynamic diagrams for another proposition.
Another_Diagram_Interpretation(premise_type)
    if premise_type == "A" then
        if subject_term == "M" then
            Create another square with diagonal which always contain the blue
              square
        else
            Create another square with diagonal which always reside inside the
              blue square
        end
    else if premise_type == "E" then
        Create another square with diagonal which always reside outside the
          blue square and always reside in U
    else if premise_type == "I" then
        if subject_term == "M" then
            Create square with diagonal which always contains the right-angle
              triangle portion of blue square
        else
        end
        if blue square does not exist then
            Create blue square using blue right-angle triangle portion
        end
        Create right angled triangle which always reside inside the blue square
```

```
    else if premise_type == "O" then
        if subject_term == "M" then
            Create square with diagonal which does not contains the
              right-angle triangle portion of blue square
        else
        if blue square does not exist then
            Create blue square using blue right-angle triangle portion
        Create right angled triangle which always reside outside the blue
          square and always reside in U
```

Algorithm 4: Validity_of_Conclusion

```
Data: Get diagrams of proposition 1 and 2
Result: Decide Validity of Conclusion
Validity_of_Conclusion()
    if "S" exists always inside "P" then
        UniversalAffirmativeProposition
    else if "S" exists always outside "P" then
        UniversalNegativeProposition
    else if right angle triangle portion of "S" exists always inside "P" then
        ParticularAffirmativeProposition
    else if right angle triangle portion of "S" exists always outside "P" then
        ParticularNegativeProposition
    else
        InvalidSyllogism
```

4 Summary and Conclusion

The usage of diagrams for syllogistic is age-old. Recently, there has been a renewed interest among logicians in using diagrammatic representation in logic. This paper reintroduces the MMR with static and dynamic diagrams. This revision is called as the RMMR, which uses static and dynamic interfaces for syllogistic thinking. It enables a user to perceive the line of reasoning and states of affairs. Moreover, we also develop algorithms for an even more user-friendly interface.

Using diagrammatic techniques in syllogistic logic helps us reason well since our faculty of cognitive reasoning is more visually inclined. Euler circles, Venn-Peirce framework, and Carroll diagrams are widely used in syllogistic. It is also true that rules of formation and transformation guide diagrammatic reasoning, and sometimes, it is difficult to comprehend these rules. An expert user or logician has ways to tackle the nitty-gritty of formations and transformations, but things seem pretty challenging for users of intermediate or beginner level. At this point, we need dynamic interfaces to help us through. In this paper, we address such and related problems for syllogistic reasoning with the help of algorithms. We also plan to extend RMMR for non-syllogistic reasoning (especially with three to four premises) as an avenue for the future.

References

1. Bauer, M.I., Johnson-Laird, P.: How Diagrams can improve Reasoning. Psychological Science **4**(6), 372-378 (1993)
2. Bosley, R.: The Geometry of Diagrams and the Logic of Syllogisms. In: Moktefi, A., Shin, S-J.(Eds.) Visual Reasoning with Diagrams, pp. 19–31. Springer, Basel (2013)
3. Carroll, L.: The Game of Logic. 1st edn Macmillan and Co., London (1886)
4. Castro-Manzano J.M.: Syllogistic with Jigsaw Puzzle Diagrams. In: Chapman P., Stapleton G., Moktefi A., Perez-Kriz S., Bellucci F. (eds) Diagrammatic Representation and Inference. Diagrams 2018. Lecture Notes in Computer Science, vol 10871. pp 657–671 Springer, Cham (2018). https://doi.org/10.1007/$978-3-319-91376-6_5 8$
5. Gursoy, K.N., Senturk, I., Oner, T., Gursoy, A.: A new algorithmic decision for categorical syllogisms via Carroll's diagrams. Soft Computing **2**(4), 11337–11346 (2020)
6. Hammer, E., Shin, S-J.: Euler's Visual Logic. History and Philosophy of Logic **19**(1), 1–29 (2007)
7. Kumova B.İ., Çakır H.: Algorithmic Decision of Syllogisms. In: García-Pedrajas N., Herrera F., Fyfe C., Benítez J.M., Ali M. (eds) Trends in Applied Intelligent Systems. IEA/AIE 2010. Lecture Notes in Computer Science, vol 6097. pp 28–38 Springer, Berlin, Heidelberg. (2010) https://doi.org/10.1007/$978-3-642-13025-0_4$
8. Larkin, J. H., Simon, H. A.: Why a diagram(sometimes) worth Ten Thousand Words. Cognitive Science **11**(1) 65-100 (1987)
9. Lemanski, J.: Logic Diagrams in the Weigel and Weise Circles. History and Philosophy of Logic **39**(1), 1–26 (2017)
10. Lemanski, J.: Oppositional Geometry in the Diagrammatic Calculus CL. South American Journal of Logic **3**(2), 517–531 (2017)
11. Lemanski, J.: Euler-type Diagrams and the Quantification. Journal of Philosophical Logic **49**, 401–416 (2020)
12. Lemon, O., De Rijke, M., Shimojima, A.: Editorial: Efficacy of Diagrammatic Reasoning: Visual Logic, Language, and Information. Journal of Logic, Language and Information **8**(3) 265–271 (1999)
13. Morita, K., Noritaka, N., Emura, E.: An efficient reasoning method for syllogism and its application to knowledge processing. Systems and Computers in Japan , **19**(3), 20–31 (1998)
14. Pagnan, R.: A Diagrammatic Calculus of Syllogisms. In: Moktefi, A., Shin, S-J.(Eds.) Visual Reasoning with Diagrams , pp. 33–53. Springer, Basel (2013)
15. Sato, Y., Mineshima, K.: How Diagrams Can Support Syllogistic Reasoning: An Experimental Study. Journal of Logic, Language, and Information, **24**(4) 409-455 (2015)

16. Sautter, F. T., Secco G.D.: A Simple Decision Method for Syllogistic. In: Chapman P., Stapleton G., Moktefi A., Perez-Kriz S., Bellucci F. (eds) Diagrammatic Representation and Inference. Diagrams 2018. Lecture Notes in Computer Science, vol 10871. pp 708–711 Springer, Cham (2018). https://doi.org/10.1007/978 $-$ 3 $-$ 319 $-$ 91376 $-$ $6_6$4
17. Sautter, F. T.: Diagramas para Juízos Infinitos. Revista Portuguesa de Filosofia **73**(3-4), 1115–1136 (2017)
18. Sautter, F. T.: A Bunch of Diagrammatic Methods for Syllogistic. Logica Universalis **13**, 21-36 (2019)
19. Sharma. S. S.: Method of Minimal Representation: An Alternative Diagrammatic Technique to test the Validity of Syllogisms. In: Stapleton, G., Howse, J., Lee, J. (eds.) 5th International Conference, Diagrams 2008, LNAI, vol. 5223, pp. 412–414. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87730-1
20. Sharma S. S.: Interpreting Squares of Oppositions with the Help of Diagrams. In: The Square of Opposition: A General Framework for Cognition, pp. 175–194. Peter Lang, (2012)
21. Sharma, S.S.: Perfect Syllogisms and the Method of Minimal Representation. In: CEUR Workshop Proceedings 1132 pp. 17–22. (2014)
22. Shin, S.-J.: The Logical Status of Diagrams. 1st edn. Cambridge University Press, New York (1994)