Solving Combinatorial Problems using Coloured Petri Nets

Somsak Vanit-Anunchai

School of Telecommunication Engineering, Institute of Engineering, Suranaree University of Technology, Muang, Nakhon Ratchasima,Thailand

Abstract

This paper explores the use of Coloured Petri Nets (CPNs) as a formal modeling tool for solving combinatorial problems. Focusing on a book distribution problem, we demonstrate how CPNs help visualize and systematically verify various proposed solutions. By constructing CPN models and generating the corresponding state spaces, we identify the correct solution and diagnose reasoning errors in alternative approaches. The study highlights the similarities between combinatorial problem-solving and system modeling, particularly in addressing hidden constraints and managing state space explosion. Our findings underscore the potential of formal methods like CPNs as educational tools for enhancing understanding in combinatorics and improving students' problem-solving skills.

Keywords

State space analysis, The inclusion-exclusion principle, The multiplication principle, Stars and bars technique.

1. Introduction

Combinatorics is the study of counting, arrangement, and selection of objects. It plays a foundational role in high school and undergraduate mathematics curricula. It introduces students to the essential concepts such as the multiplication rule, permutations, and combinations, which frequently arise in real-world contexts like selecting outfits or calculating poker odds. Despite its significance, many students struggle with combinatorial reasoning, often due to a lack of conceptual understanding and an over-reliance on memorized formulas.

Drawing from real educational experiences, a student approached me with a distribution problem after attending the first-round POSN[1] camp. "How can 10 distinct books and 8 identical red balls be distributed among 3 students, ensuring that each student receives at least one book and one ball?" Multiple solution methods proposed at the camp yielded differing answers. The student asked two critical questions: (1) Which answer is correct? and (2) Why or how are the other methods incorrect?

To address the first question, I suggested using CPN Tools [1] to model the problem, as the Coloured Petri Net (CPN) framework [2, 3] naturally represents token distributions specified by the problem. By generating the state space, the number of terminal markings would reveal the correct solution. For the second question, I recommended constructing separate models for each proposed method. By systematically exploring and comparing their respective state spaces, we could identify the reasoning errors leading to incorrect answers. During this modeling exercise, we encountered the well-known state space explosion problem. To manage this complexity, I advised simplifying the model by considering only 6 distinct books and temporarily omitting the red balls. Once the problem-solving methods are well understood on this simplified model, they can be directly extended to larger, more complex problems.

This suggestion illuminated similarities between combinatorial reasoning and formal verification in system modeling. In combinatorial problem, constraints are not always explicitly stated. Hidden or implicit conditions often require careful reading and thoughtful interpretation. These *ambiguities* are analogous to those encountered in system modeling, where informal requirements may lack precision.

PeNGE'25: Petri Net Games, Examples and Quizzes for Education, Contest and Fun. June 24, 2025, Paris, France

somsav@sut.ac.th (S. Vanit-Anunchai)
0000-0003-4972-7713 (S. Vanit-Anunchai)

[1]The Promotion of Academic Olympiad and Development of Science Education Foundation (POSN) is a Thai government-supported organization aimed at enhancing science and mathematics education, preparing students for international competitions.

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

Such uncertainties can be effectively addressed through formal specification techniques, which help to make assumptions more explicit. In combinatorial analysis, enumerating all possible configurations quickly leads to an exponential growth of the solution space. The phenomenon closely mirrors the *state space explosion* encountered in formal methods. In both contexts, abstraction, through simplification or reduction, becomes crucial to making problems tractable. Furthermore, the systematic exploration of state spaces serves not only to identify correct solutions but also to provide *critical insights* into why alternative methods fail. This diagnostic process reveals underlying reasoning errors and uncovers overlooked constraints.

This paper investigates the relationship between combinatorial problem-solving and formal methods. In particular, we use Coloured Petri Nets (CPNs) to systematically model and analyze such problems. The remainder of the paper is organized as follows: Section 2 discusses the CPN models for the book distribution problem, identifying the correct methods and analyzing the reasoning errors in incorrect approaches. Section 3 reviews related work, and Section 4 concludes the paper.

2. The CPN Model of the Book Distribution Problem

To solve the book distribution problem, we divide the problem into two distinct stages: first, the distribution of 10 distinct books, and second, the distribution of 8 identical balls. According to the Multiplication Principle[2], the total number of possible outcomes is the product of the number of ways each event can occur. Since the distribution of identical balls can be addressed separately using the Stars and Bars[3] technique, this section focuses primarily on modeling the distribution of the distinct books. Without loss of generality, we reduce the number of books to 6 to avoid state space explosion.

Figure 1 presents the Coloured Petri Nets (CPNs) model, a simplified version of the original problem, which captures the distribution of 6 distinct books among 3 students. The goal is to ensure that each student receives at least one book.

[2]The Multiplication Principle: If one event can occur in m ways and a second event can occur in n ways, then the sequence of these two events can occur in $m \times n$ ways.

[3]Stars and bars is a combinatorial technique for counting ways to divide identical items into distinct groups. To find positive integer solutions to $x_1 + x_2 + \cdots + x_k = n$, we place $k-1$ bars among n stars. The number of ways is $\binom{n-1}{k-1}$.

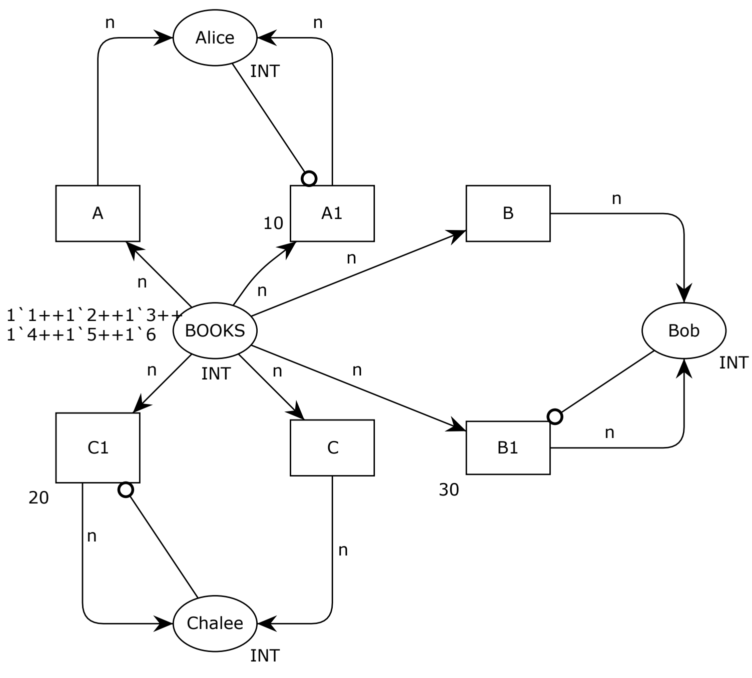

Figure 1: The CPN model of the book distribution problem.

Depicted as an oval, the place *BOOKS* initially contains tokens representing the 6 distinct books (1 through 6), each treated as a unique value of type INT. The places *Alice, Bob, Chalee* represent where books are assigned to the respective students. Transitions shown as rectangles, represent the possible actions of assigning books from the place *BOOKS* to each of the students. Transitions labeled *A1, B1, and C1* are used in conjunction with inhibitor arcs, which prevent firing unless the respective student's place is empty. These three transitions have higher priorities (10, 20 and 30 respectively) than transitions *A, B, and C* to ensure that each student receives one book in the first round. The total state space consists of 2,137 states. There are 540 terminal markings. Each represents a valid distribution where all books are assigned and every student has at least one book.

2.1. The First Proposed Method

To compute the number of valid distributions using the inclusion-exclusion principle[4] , note that each book has 3 possible recipients, giving a total of 3^6 = 729 ways. However, this count includes cases where one or two children receive no books. To correct this overlap:

- Subtract the number of distributions where only 2 children receive books: 3×2^6 ways,
- Add back the cases where only 1 child receives all books: 3 ways.

Thus, the total number of valid distributions where each child receives at least one book is:

$$3^6 - 3 \times 2^6 + 3 = 540$$

[4]The inclusion-exclusion principle is used to count elements in the union of overlapping sets. For two sets, the formula is:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

It corrects for elements counted twice in the overlap. This idea extends to three or more sets by alternately subtracting and adding intersections.

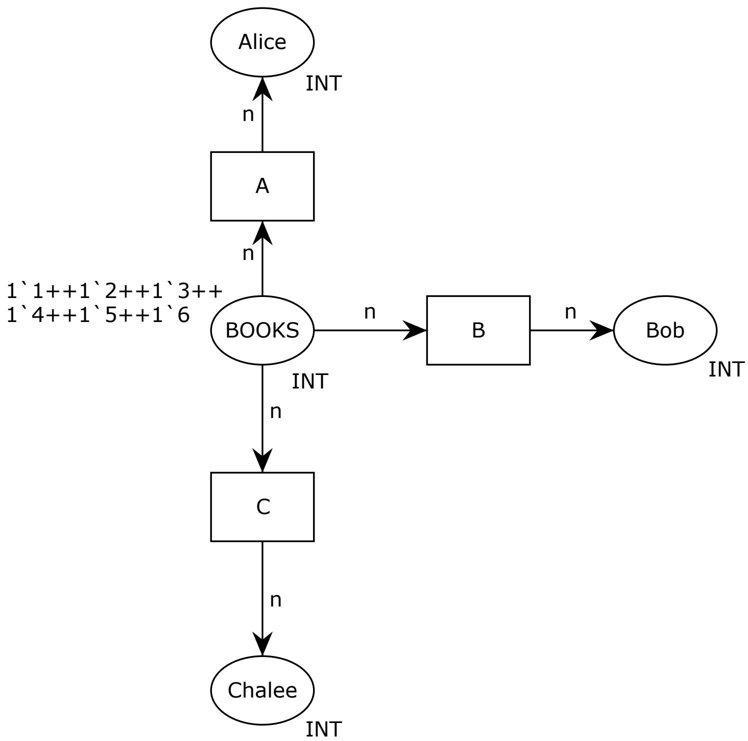

Figure 2: The CPN model of the first method distributing books across all possible scenarios.

Figure 2 illustrates the CPN model constructed to distribute books across all possible scenarios (729 configurations), including those in which some children do not receive any books. The inclusion-exclusion principle is applied using the CPN query language shown in Fig. 3. Determining the number of states in which Alice does not receive a book yields 64 states (Fig. 4). This count also includes a case where both Alice and Bob receive no books. Owing to the symmetry of the model, the same result applies to Bob and Chalee. When subtracting the 64 states where Alice does not receive a book and the 64 states where Bob does not receive a book, the states in which both Alice and Bob receive no books are subtracted twice. Consequently, these states must be added back in the final computation to correctly apply the inclusion-exclusion principle.

```
fun ev1(n) = if (Mark.BOOK'Alice 1 n)<>empty andalso (Mark.BOOK'Bob 1 n)<>empty
                andalso (Mark.BOOK'Chalee 1 n)<>empty then true else false;
fun ev2(n) = if (Mark.BOOK'Alice 1 n) = empty then true else false;
fun ev3(n) = if (Mark.BOOK'Alice 1 n) = empty
                 andalso (Mark.BOOK'Bob 1 n)=  empty then true else false;
val _ =  print("Satifies  Some students have not received any books.:");-
length(ListDeadMarkings());
val _ = print("Satifies  Alice has not received any books.:");
length(PredNodes(ListDeadMarkings(), ev2, NoLimit));
val _ = print("Satifies  Neither Alice nor Bob has received any books.:");-
length( PredNodes(ListDeadMarkings(), ev3, NoLimit));
val _ = print("Satifies  Each student receives at least one book.:");-
length (PredNodes(ListDeadMarkings(), ev1, NoLimit));
```

Figure 3: The ML code used to implement the inclusion-exclusion query.

```
Satifies - Some students have not received any books.:val it = 729 : int
Satifies - Alice has not received any books.:val it = 64 : int
Satifies - Neither Alice nor Bob has received any books.:val it = 1 : int
Satifies - Each student receives at least one book.:val it = 540 : int
```

Figure 4: Query result for the inclusion-exclusion method.

2.2. The Second Proposed Method

The second approach organizes the arrangement in two steps. First, arrange the six distinct books in a row, which can be done in 6!= 720 ways. Then, in each case, use the Stars and Bars method to divide the six books into three distinct groups, which can be done in $\binom{6-1}{3-1}$= 10 ways. Thus, the total number of possible arrangements by this method is $720 \times 10 = 7,200$ ways.

The second method leads to an incorrect answer because it allows the books within each group to be permuted, whereas in the actual problem, the books held by each child are not reordered. In fact, the second method models a scenario where books are arranged onto three shelves, with the order of books on each shelf being significant.

Figure 5 illustrates the CPN model for the case where books are arranged onto three shelves with internal permutations allowed. The model closely resembles that in Fig. 1 but to capture the ordering of books, tokens are represented by the List type instead of a multi-set. Transitions *A1, B1, C1* play the same purpose as the corresponding transitions in Fig. 1. However, rather than using inhibitor arcs, a guard [length(ln) = 0] is employed to check whether the token is a null list. The generated state space contains 12,757 states and 7,200 terminal markings.

2.3. The Third Proposed Method

This method divides the distribution process into three stages. First, select a group of three distinct books from the six available, which can be done in $\binom{6}{3}$ =20 ways. Second, distribute these three books in order among the three children, which can be done in 3! = 6 ways. Finally, assign each of the remaining

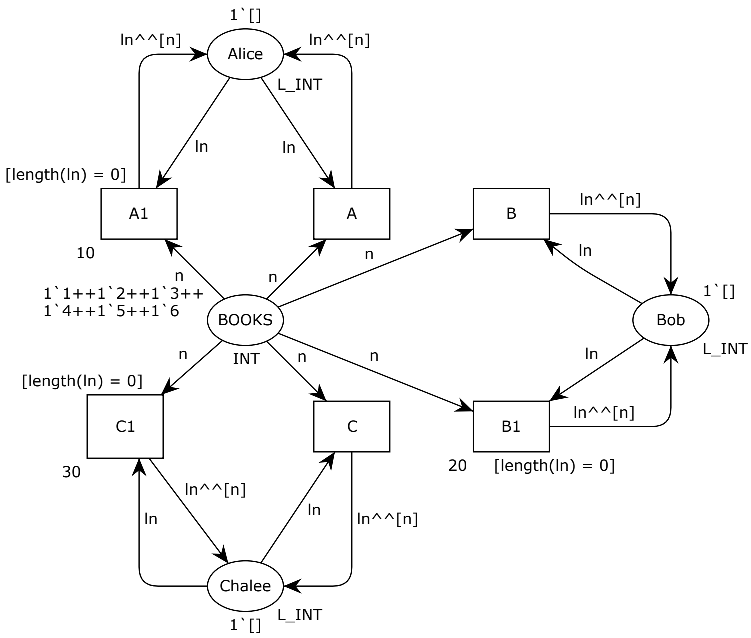

Figure 5: The CPN model of the second method: book distribution onto three shelves.

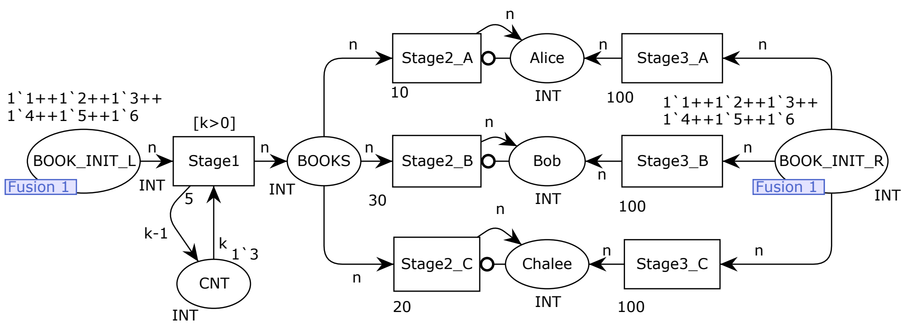

Figure 6: The CPN model for the third method dividing the distribution process into three stages.

three books independently to any of the three children, giving 3^3 =27 possible distributions. Thus, the total number of arrangements according to this method is $20 \times 6 \times 27 = 3,240$ ways.

Figure 6 shows the CPN diagram representing the distribution process in three stages. The resulting state space comprises 2,322 states and 540 terminal markings, which correspond to the correct number of valid distributions. When transitions in the second and third stages are disabled, the state space yields 20 terminal markings matching the number of combinations from the first stage. Disabling only the third stage results in 120 terminal markings, consistent with the product of the first two stages. When all three stages are active, the number of terminal markings is 540, indicating that the original 3,240 configurations have been folded together due to equivalence in the final distributions.

A counterexample is taken from the generated state space. This confirms that the method overcounts distinct arrangements. Consider the following two cases:

- Case I: Alice selects Book No.1 .
- Case II: Alice selects Book No.4

Both cases then proceed with identical actions. In the final step, Case I: Alice selects Book No.4, and Case II: Alice selects Book No.1. This leads to the same final distribution. However, the multiplication principle treats them as distinct due to different initial choices, even though the outcomes are equivalent. Therefore, this method results in overcounting and produces an incorrect answer.

2.4. The Fourth Proposed Method

This final approach presumes that the final distribution of books among the three children falls into one of the following patterns: (4,1,1) (3,2,1), or (2,2,2). The number of possible scenarios for each pattern is:

- 3 scenarios for (4,1,1),
- 6 scenarios for (3,2,1),
- 1 scenarios for (2,2,2).

- For the $(4,1,1)$ distribution:

$$\binom{6}{4} \times \binom{2}{1} \times 1 \times 3 = 30 \times 3 = 90 \text{ ways.}$$

- For the $(3,2,1)$ distribution:

$$\binom{6}{3} \times \binom{3}{2} \times 1 \times 6 = 60 \times 6 = 360 \text{ ways.}$$

- For the $(2,2,2)$ distribution:

$$\binom{6}{2} \times \binom{4}{2} \times \binom{2}{2} \times 1 = 90 \text{ ways.}$$

Thus, the total number of possible arrangements is: 90+360+90 = 540 ways.

Figure 7 illustrates the CPN model for the configuartion (4,1,1). The model closely resembles that in Fig. 2 but the number of books assigned to each student is bounded by the value in places *CNT_A, CNT_B and CNT_C*. Each time a student receives one book, the corresponding counter is decremented by one. The guard condition [k > 0] ensures that no students receives more books than they are allocated. The generated state space contains 909 states and 30 terminal markings. The other two configuartions (3,2,1) and (2,2,2) can be easily obtained by setting tokens in Places *CNT_A, CNT_B, and CNT_C*.

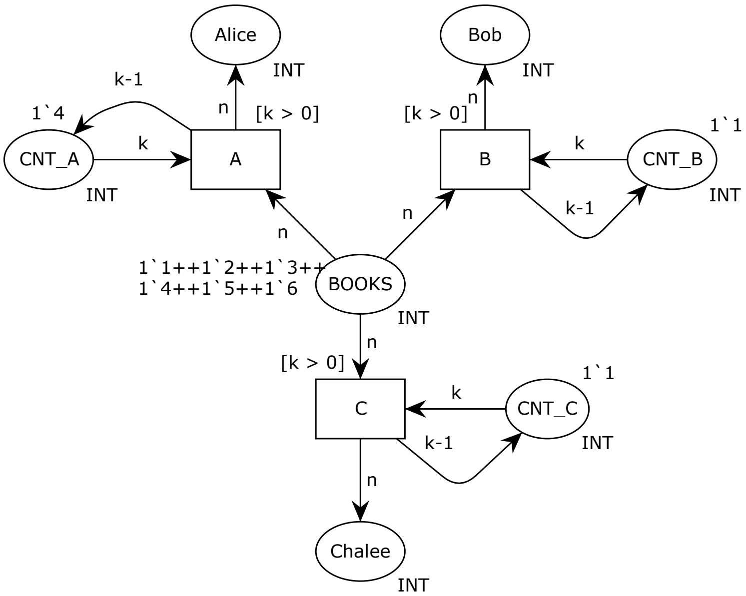

Figure 7: The CPN model for the fourth method.

3. Related work

Combinatorics education has been widely discussed in research, particularly with regard to the difficulties when the students learn the subject. A recurring theme in the literature is the over-reliance on formulas and a lack of conceptual understanding among students, which hinders their ability to grasp the principles of counting and combinatorial reasoning.

Syahputra [4] investigates students' difficulties in solving combinatorics problems and their strategies. The study involving 36 high school students and 67 first-year college mathematics education students revealed very poor combinatorial ability. Given five problems, only 35% answered the first correctly, 10.68% the second, none got the third right, 1.9% solved the fourth, and only 0.97% solved the fifth. The analysis suggests that most students did not understand the problems, rarely used enumeration, and generally avoided building mathematical models but relying on memorized formulas. Intensive practice using enumeration, pattern recognition, and trial-and-error methods is recommended to improve their skills.

Lockwood [5] shows students struggle with counting problems due to incorrect understanding of combinatorics. Lockwood developed a model to analyze students thinking, focusing on relationships among counting processes, outcomes, and formulas. The model describes how students conceptualize counting tasks and explains common difficulties. It also offers a framework for teachers designing experiments and identifying sources of student errors. Finally, the model provides a foundation for better instructional strategies in combinatorics education.

Sriraman and English [6] explains that students often struggle with combinatorics due to confusion between permutations and combinations. It defines combinatorics as the art of counting arrangements of finite sets and emphasizes its role in flexible and independent thinking. Researchers use multiple representations for students to build structural understanding. Collaborative problem-solving and problem-posing activities can further enhance creativity, and conceptual understanding.

4. Conclusion

This paper explored the application of Coloured Petri Nets to model a combinatorial problem and evaluate potential solutions. Using CPNs to visualize and analyze the problem, we can identify the correct solution and diagnose the reasoning errors such as overcounting and unintended permutations. Through abstraction and model simplification, we can reduce the complexity and make the problem tractable. Using formal methods like CPNs provides better understanding and teaching combinatorial concepts. Finally, this study underscores the value of formal tools in both system modeling and mathematical problem-solving education.

Declaration on Generative AI

During the preparation of this work, the author used GPT-4 in order to: Grammar and spelling check. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] CPN Tools, CPN Tools home page, http://cpntools.org, 2018.

[2] K. Jensen, L. Kristensen, Coloured Petri Nets: Modelling and Validation of Concurrent Systems, Springer, Heidelberg, 2009.

[3] K. Jensen, L. M. Kristensen, Colored petri nets: a graphical language for formal modeling and validation of concurrent systems, Commun. ACM 58 (2015) 61–70. URL: http://doi.acm.org/10.1145/2663340. doi:10.1145/2663340.

[4] E. Syahputra, Combinatorial Thinking (Analysis of Student Difficulties and Alernative Solution), in: The Third International Seminar On Trends In Science and Science Education, 7-8 October 2016, Medan, Indonesia, 2016.

[5] E. Lockwood, A model of students combinatorial thinking, The Journal of Mathematical Behavior 32 (2013) 251–265. URL: https://www.sciencedirect.com/science/article/pii/S0732312313000230. doi:https://doi.org/10.1016/j.jmathb.2013.02.008.

[6] B. Sriraman, L. D. English, Connecting research to teaching: Combinatorial mathematics: Research into practice, Mathematics Teacher 98 (2004) 182–187.