Constraint-based Student Modeling in Probability Story Problems with Scaffolding Techniques

https://doi.org/10.3991/ijet.v13i01.7397

Nabila Khodeir (✉), Nayer Wanas
Electronics Research Institute, Cairo, Egypt
nkhodeir@eri.sci.eg

Hanan Elazhary
King Abdulaziz University, Jeddah, Saudi Arabia
Electronics Research Institute, Cairo, Egypt

Abstract—Constraint-Based student Modeling (CBM) is an important technique employed in intelligent tutoring systems to model student knowledge to provide relevant assistance. This paper introduces the Math Story Problem Tutor (MAST), a Web-based intelligent tutoring system for probability story problems, which is able to generate problems of different contexts, types and difficulty levels for self-paced learning. Constraints in MAST are specified at a low-level of granularity to allow fine-grained diagnosis of the student error. Furthermore, MAST extends CBM to address errors due to misunderstanding of the narrative story. It can locate and highlight keywords that may have been overlooked or misunderstood leading to an error. This is achieved by utilizing the roles and syntaxes of the problem segments and the semantic descriptions of the keywords that are defined through the Natural Language Generation (NLG) methods deployed in the story problem generation. MAST also integrates CBM with scaffolding questions and feedback to provide various forms of help and guidance to the student. This allows the student to discover and correct any errors in his/her solution. The tutoring effect of MAST has been evaluated empirically using different tests (paired samples t-test, normalized knowledge gain, Mann-Whitney U test, and power learning curves). The results demonstrated a positive effect of MAST on improving the exam scores, and the normalized knowledge gain. Additionally, keyword highlighting of errors integrated in MAST has been shown to speed up learning and improve reduction of average percentage of violated constraints along the learning curve. This suggests that using various forms of assistance can help speedup learning, and that well-designed tutoring systems providing relevant help can be superior to traditional approaches.

Keywords—Constraint-based Modeling, Intelligent Tutoring Systems, Scaffolding Feedback, Scaffolding Questions.

1 Introduction

Intelligent Tutoring Systems (ITSs) are computerized tutoring systems that typically incorporate AI techniques within their operation. Such techniques support simulating tutoring activities such as student coaching, student guidance, and misconception diagnosis. In turn, ITSs have the potential to involve the student in an active learning environment matching the effective learning outcome of an individual tutor. ITSs typically provide adaptable support to each student based on modeling of the student knowledge and his/her characteristics. The student model may allow individualization of the system interactions, such as providing student-specific feedback or selecting the next problem to solve.

The student model is able to represent one or more of the student features [1] such as knowledge [2, 3], interests [4, 5], goals [6, 7] and individual traits [8, 9, 10]. The student knowledge model stores the student's cognitive state with respect to a specific subject domain. It is based on analysis of behaviors that are related to the problem-solving process such as the student answer, the student errors and the number of trials to reach the correct answer.

Constraint-Based Modeling (CBM) is a student knowledge modeling technique which is based on Ohlsson's theory of learning from performance errors [11]. This translates to representing the fundamental principles of the domain as a set of constraints, where constraints define equivalence classes of problem states. Each constraint has a relevance condition, at which it is applicable and a satisfaction condition, whose violation indicates an error [11, 12, 13]. Using CBM allows the diagnosis of the student errors, without an explicit or generative model of buggy versions of a skill. The student's knowledge is described in terms of both the satisfied and violated constraints. The tutoring system reacts to such violation by advising the student on the error(s) using the information contained in the violated constraints [14].

CBM has been employed in different domains such as Structured Query Language (SQL) of relational database systems [15], database modeling [16, 17, 18], data normalization [19, 20] and object-oriented analysis and design using Unified Modeling Language (UML) [21, 22]. Although CBM is typically employed with word problems in different domains, it is limited to checking the student answer syntactically and semantically based on defined constraints. It generally falls short of taking into consideration the possibility of the student error due to misinterpretation of the story problem. An objective of this work is extending CBM to address the student misinterpretations errors.

This paper presents the Math Story Problem Tutor (MAST), a Web-based intelligent tutoring system for probability story problems. MAST has the ability to generate problems of different contexts, types and difficulty levels for the purpose of facilitating self-paced learning. The paper focuses on CBM in MAST aiming to diagnose the student errors, and thus effectively tutor him/her. The paper provides answers to the following research questions:

- If story problems are automatically generated using NLG, and the role of each problem segment and its syntax are known in advance, could that affect the capability of the system to diagnose the students' misinterpretation errors?
- Could we relate misinterpretation errors to corresponding keywords within specific problem segments?
- Could the ability to locate the corresponding keywords of misinterpretation errors be used as a type of help and guiding to the student?
- Could we use scaffolding technology to help students to answer the main question?

The rest of the paper is organized as follows: First, we discuss the application of CBM in story problems and the different scaffolding approaches to highlight the contributions of the paper along these two dimensions. Second, we introduce MAST, explaining the structure of a typical problem and how it facilitates self-paced learning. This is followed by a discussion of the different constraint types in MAST and how they are exploited in CBM. We also explain how the different scaffolding techniques are integrated with CBM to provide several forms of assistance. Finally, the results of the empirical evaluation of MAST and the effectiveness of its student modeling approach are presented followed by the conclusion and possible future research.

2 Constraint-Based Student Modeling in Story Problems

The goal of student modeling is to diagnose the student's misunderstanding of a given subject and react accordingly. Student modeling techniques can be broadly classified into (i) short-term and (ii) long-term modeling techniques. In short-term student modeling, errors in the student's solution to a given problem are diagnosed to provide relevant help [13]. Long-term student modeling, on the other hand, considers the student interactions with the system over time to build a more accurate model. Long-term modeling allows adding an adaptation dimension to the tutoring system such as selection of an appropriate following problem [23]. Representation of the student model, especially the knowledge model, varies from simple numeric rankings to complex networks.

Techniques for student modeling generally rely on either (i) Artificial Intelligence (AI) methods or (ii) cognitive science approaches. AI techniques include machine learning agents [24], fuzzy logic, Bayesian networks [25] and Artificial Neural Networks (ANNs) [26] that reason about the student knowledge. They may be used for either long-term or short-term student modeling. Cognitive science techniques, on the other hand, are based on the assumption that human knowledge (and errors) can be modeled. Two prominent examples of cognitive science techniques are model-tracing [27] and CBM [13] that are widely used for short-term student modeling.

In model tracing, traditional If-Then production rules are used to encode the knowledge of human experts and how they solve problems according to both the given facts and goal. An error is detected when the student answer, or answer step, does not match any production rule. The tutoring system typically reacts by advising the student on the correct production rule to be applied. Student bugs may also be augmented as buggy rules to allow advising the student about the error [28].

Ohlsson proposed CBM as an approach to avoid some limitations of model-tracing, such as having runnable models of the expert and the student in addition to identifying and augmenting the student bugs. As previously noted, CBM is based on capturing the domain principles using a set of constraints. A constraint is typically represented by the tuple <Cr, Cs>, where the relevance condition Cr is the condition under which the constraint is applicable, while the satisfaction condition Cs is the condition that should be satisfied such that the constraint is not violated [13]. Identifying errors, in this case, is based on tracking the violated constraints. In such a situation, the tutoring system can respond by giving advice to the student about the error, even without being able to replicate it.

In spite of employing CBM in story problems in numerous domains, to the best of our knowledge, none of the existing research studies in the literature takes into consideration the possibility of the student error due to misinterpretation of the story problem itself. This is mainly due to the complexity of this task that requires the natural language analysis of the problem story. MAST addresses this shortcoming, leveraging the automated natural language generation of its word problems and the fact that the syntax and role of each problem segment is known in advance. Accordingly, MAST extends the classical representation of a constraint to be able to locate keywords that may not have been taken into consideration by the student and integrates CBM with scaffolding techniques to provide various forms of assistance. The paper also introduces new types of constraints specified at a low-level of granularity to allow fine-grained diagnosis of the student error and hence more specific feedback as explained in the following sections.

3 Scaffolding

Scaffolding is a type of support provided during the learning process. This support is intended to be individualized to the specific requirements of the student with the objective of helping the student achieve his/her learning goals [29, 30]. Moreover, it is designed to stimulate a deeper level of learning.

There are two prominent scaffolding techniques (i) scaffolding questions [31, 32, 33, 34, 35] and (ii) scaffolding feedback [36]. Scaffolding questions are intended to be a sequence of questions to aid the students build understanding. These series of questions are called "micro plans" [37]. Heffernan et. al embraced the term scaffolding question to refer to what the tutor asks as a way to help the student in solving of the original problem [31]. In scaffolding questions [31, 32], the problem is broken into sub-problems [33]. Harrington and Holroyd [35] showed that scaffolding questions have a significant impact on tutoring the students in comparison to merely providing hints. A variant of this approach has been proposed in which each scaffolding question is provided in the form of a multiple-choice question. This question is designed, such that self-explanation is prompted in an attempt to teach the student how to solve a similar problem correctly [30].

In scaffolding feedback [36], on the other hand, the student is provided with a set of incremental hints until he/she is able to solve the problem correctly. Scaffolding feedback has been tested against three other feedback techniques (i) standard corrective feedback, where the correct answer is provided immediately to the student, (ii) minimal feedback, where the student is merely informed about whether or not the provided answer is correct, and (iii) answer-until-correct multiple-choice feedback, where the student is given minimal feedback about each answer and allowed to make other choices until the correct answer is selected. It has been shown that scaffolding feedback gives relatively better results over longer periods of time [35].

To provide extensive assistance to the student, MAST exploits both scaffolding questions and scaffolding feedback. Moreover, it extends scaffolding feedback by providing visual hints in the form of highlighted problem keywords upon detecting a student error to address the possibility of the misinterpretation of the story problem as explained in the following sections.

4 The Math Story Problem Tutor (MAST)

MAST is a Web-based intelligent tutoring system for probability story problems. Probability is favorable outcome over all possible outcomes. For example, the probability to get 3 when we roll a die is 1/6, where there are six possible outcomes (1, 2, 3, 4, 5, 6) and 3 is one of them. In the probability story or word problem, the information is taken from the context of the problem. MAST can generate problems of different contexts. Problem types within a given context are classified into several categories depending on the number of elements, and organization of the heading stories. Difficulty of heading stories is proportional to the level of complexity of the structure used. As an illustrative example, within the context of tossing a coin and rolling a die, problem types are classified into four categories:

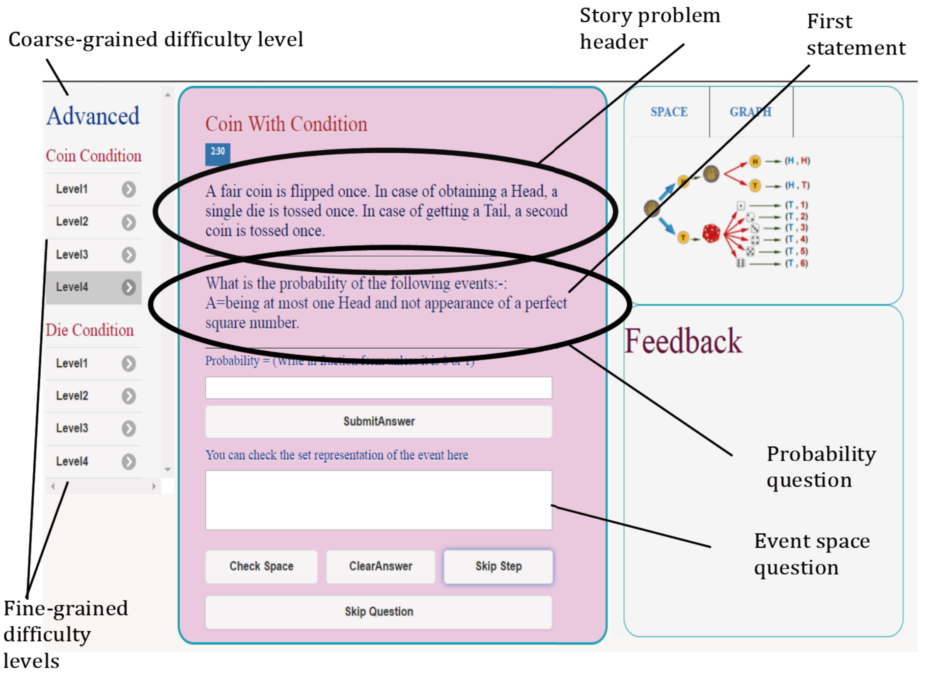
- Low difficulty level: problems involving a single object such as a single die or a single coin.
- Medium difficulty level: problems involving more than one object of the same type such as two dice.
- High difficulty level: problems involving more than one object of different types such as a coin and a die.
- Advanced difficulty level: problems involving conditions. For example, a single coin is tossed and according to the outcome, another coin may be tossed or a die may be rolled.

It is worth mentioning that the difficulty levels of these problems categories have been quantified by engaging domain experts. A questionnaire with a broad set of problems was addressed by these domain experts where they rated the questions on a Likert scale between 1 (extremely easy) and 9 (extremely difficult). The average of the difficulty level assigned to each question type was used, as we also evaluated the reliability of the difficulty levels using Intra-Class Correlation (ICC). The different question types were then classified into the four categories.

Problems of a given type are further classified into fine-grained difficulty levels according to the operations involved in their questions. Accordingly, MAST allows the student to select problems of different contexts, types and difficulty levels for self-paced learning. Figure 1 presents an example problem within the interface of MAST.

4.1 Basic Problem Structure in MAST

A typical MAST problem is composed of a header followed by one or more probability questions presented to the student one at a time. The header of each MAST problem includes a probability story describing a situation. This could involve one or more objects, and one or more operations depending on the difficulty level of the story problem. On the other hand, a *probability question* asks the student to provide the probability of a specific event. An *event space question* is an optional question, which asks for the student to list the set of elements that represent the event.



A fair coin is flipped once. In case of obtaining a Head, a single die is tossed once. In case of getting a Tail, a second coin is tossed once.

What is the probability of the following events:-:
A=being at most one Head and not appearance of a perfect square number.

Probability = (Write in fraction form unless it is 0 or 1)

You can check the set representation of the event here

Fig. 1. A screenshot of MAST depicting fine-grained difficulty levels within the coarse-grained Advanced difficulty level.

4.2 Probability Question Structure

A probability question is formed of canned text followed by a statement including a number of operations. Three examples of probability questions canned text are "Calculate the probability of the following events," "Calculate the probability of each of the following events" and "What is the probability of the following events".

As shown in the Figure 1, in addition to the canned text, each probability question includes a statement with zero or more operations. For example, the statement in the probability question “occurrence of a head” includes no operations. On the other hand, the statement in the probability question “being at most one Head and not appearance of a perfect square number” includes four operations: "at most", "and", “not” and " perfect square number ". It is worth noting that each of those operations has a difficulty level. The difficulty level of a given probability question is the aggregation of the difficulty levels of the operations within its statement. The difficulty level of a given MAST problem is equal to the difficulty level of each of its probability questions. This implies that all the probability questions that appear within a given probability problem are all of the same difficulty level.

The generation process of the different statements is based on the selection of one or more operations from the specified operations for each object type. Instantiation of associated variables values (if it is required for the operation) are restricted by some rules to ensure that semantic consistency is achieved.

4.3 Optional Scaffolding MAST Questions

Solving the probability question of a specific event requires the student to estimate (i) the sample space of the problem based on the problem header and (ii) the set representation of the event in the question statements. For example, if the story problem is flipping one die and the probability question is asking about the probability of a perfect square number, the student has to know the sample space which is {1, 2, 3, 4, 5, 6} and the set representation of the event that represents the perfect square numbers in the sample space which is {1, 4}. In turn, the student can estimate the probability, which is the number of the event space elements divided by the number of the sample space elements which is 1/3. The student can submit the probability value which is checked by the system.

MAST supports the student by providing two optional scaffolding questions to check his/her answer regarding the sample space and the set representation of the event. The first optional question is a multiple-choice sample space graph question that prompts the student to select a sample space graph relevant to the story in the problem header. Moreover, the relevant graph is displayed to the student throughout the rest of the problem to help in solving the following probability questions as illustrated in Figure 1. The second optional question focuses on the set representation of the event provided with each probability question to allow the student to write down the event set representation used in the computation and check its validity as shown in the figure. This question is formed of canned text “You can check the set representation of the event here”. Each question type has different types of constraints that are used to check the student answer and diagnose the errors. Next section will explore these different constraint types.

5 Constraint Types in MAST

As previously noted, a constraint is typically represented by the tuple <Cr, Cs>. MAST extends this classical representation of a constraint to be able to address errors due to misinterpretation of the story problem by locating keywords that may not have been taken into consideration by the student. Each constraint in MAST is represented by the following tuple:

<Cr, Cs, WSD, PSD, H, [axiom]>

Where *WSD* is the word semantic description of the keywords that should be taken into consideration by the student to satisfy the constraint, while *PSD* is the problem segment description in which the keywords are located. Each constraint is accompanied by one or more hints *H* to guide the student to satisfy the constraint in case of an error. Moreover some constraints may be accompanied by a follow-up axiom. As previously noted, MAST generates its problems using NLG methods and hence, the role or description of each problem segment is known in advance. Additionally, the syntax of each segment with semantic description of each word is defined. This makes MAST capable of locating the keywords inside the corresponding segments using merely their semantic descriptions. Subsequently, they are automatically highlighted in case of violation of the specified constraint.

It is worth mentioning that, highlighting keywords is used in KERMIT systems for the purpose of referring the student solution errors in case of misunderstanding the problem [16]. KERMIT asks the student to highlight the word or phrase modeled by each object in the ER diagram. Then, the system uses a one-to-one mapping of words of the problem text to the objects of its ideal solution to identify the corresponding student solution objects. KERMIT exploits the nature of the domain where the student solution object names are directly presented as words in the story problem. On the other hand, in other domains, such as probability in MAST, a few words in the problem have impact on the student solution. For example, in a problem talking about a coin that is tossed twice, the student answer cannot contain numeric numbers in the sample space or event set representations since the allowed elements are H (head) and T (tail) only. Therefore, if the student answer contains number(s) such as five, the word *coin* will be highlighted automatically to draw the student attention for the object in the problem. Furthermore, the word *coin* will not be mentioned in the constraint, but the object name will be used instead since it is the semantic description of this word. Using the semantic description of the word minimizes the number of constraints that have to be defined for the same error. For example, the same constraint will be used if the problem is changed to a die tossed two times and the student answer contains the symbol H or T. Some additional examples are provided in the following subsections.

In the literature, the constraints are typically classified into (i) syntax constraints and (ii) semantic constraints [38]. A syntax constraint is a problem independent general domain constraint. On the other hand, a semantic constraint relates the student answer to the correct answer. This paper classifies the syntax constraints into two types (i) domain constraints, which are domain general constraints, and (ii) context constraints, which are domain constraints in specific problem contexts. In addition, the paper introduces a special type of semantic constrains on multiple choice sample space graph questions. Accordingly, MAST constraints are classified into (i) syntax constraints which imply domain constraints and context constraints, (ii) semantic constraints and (iv) sample space graph constraints.

It is worth mentioning that, due to the NLP capability of MAST, most of the constraints take a generalized form of the error types and accordingly there are small numbers of constraints in each type. We have five domain constraints, six context constraints, 23 semantic constraints and five sample space graph constraints.

5.1 Syntax Constraints

Syntax constraints are classified into domain and context constraints. Domain constrains are problem independent and focus on the answer form according to the type of the asked question. On the other hand, the context constraints check the student consideration of the context of the problem story such as considering all problem-story mentioned objects.

Problem	Correct Probability	Constraint Parameters
A fair die is rolled once, and the number on the upper face is observed. **What is the probability of obtaining an odd number?**	1/2	***Relevance condition:*** *if the problem asks about the probability of an event in the probability question* ***Satisfaction condition:*** *the student answer must be greater than or equal to zero and less than or equal to one* ***Related keyword:*** *probability* ***Problem segment description:*** *canned text of probability question* ***Hint:*** *The probability is a number greater than or equal to zero and less than or equal to one.*
	Student probability	
	2	
(a)	**(b)**	**(c)**

Fig. 2. (a) An example MAST problem, (b) the correct and student wrong answer, and (c) an applicable syntax constraint.

5.2 Domain Constraints

The student answer can be a response to the probability question (the main question) or to the set representation of the event question (scaffolding questions). Therefore, the domain constraints deal with the possible student answer forms. For example, syntax constraints that are concerned with the probability questions ensure that the probability is provided in proper fraction form and has a numeric value that is larger than or equal to zero and smaller than or equal to one. Figure 2 shows an example MAST problem as well as an applicable syntax constraint that deals with probability questions. On the other hand, a syntax constraint that deals with event set representation questions ensure that the student answer has a correct form of a set. For example, a syntax constraint checks that the student answer elements are included between braces and the number of opened and closed parentheses match.

It is worth noting that the keywords in such constraints are always located in the corresponding question canned text and not in the problem header. This is due to the nature of the syntax constraints that are independent of the problem details. Additionally, the keyword(s) are always those that specify the basic requirements of the question and are put in the constraints directly. To illustrate, the keyword of the constraint shown in Figure 2(c) is the word "probability", since the basic requirement of the corresponding probability question is to compute the probability of an event. The student has to take this keyword into consideration and therefore it is automatically highlighted when the student gives a wrong answer that violates the constraints in Figure 2(c). Similarly, the keywords of a constraint on the event set representation questions would be the "set representation of the event" that the student has to take into consideration to preserve the general properties of the representation.

Problem	**Correct Event Set Representation**	**Constraint Parameters**

Assume tossing a fair *coin* once, and then flipping a single *coin* once in case of appearance of a Head. In case of getting a Tail, assume throwing a fair *die* once.
Calculate the probability of being exactly one Head and being a number equal 3.
You can check the set representation of the event here.
	{ }
	Student Event Set Representation
	{(H, 3)}
		Relevance condition: *if the problem asks about the event set representation*
		Satisfaction condition: *the student answer must contain only elements belonging to the problem sample space*
		Word semantic description: *object name(s)*
		Problem segment description: *conditions and results segments*
		Hint: *Not all elements are within the problem sample space.*

(a)	**(b)**	**(c)**

Fig. 3. (a) An example MAST problem, (b) the correct and student wrong answer and (c) an applicable context constraint.

5.3 Context Constraints

A context constraint is a syntax constraint in a specific problem context. Unlike the domain constraints, a context constraint needs some information from the problem header. It only checks the student answer to the set representation of the event question. For example, there is a context constraint which ensures that in case the problem header includes a single object, the student answer to the event set representation question does not include any elements of unmentioned object. Figure 3 shows a more complicated case of Advanced difficulty level problems that involve conditions resulting in excluding some elements from the spaces of the involved objects. To illustrate, in the shown example, in case of obtaining a Head, a coin would be flipped and therefore no elements from the space of the die would ever appear with a Head. So, in case, for example, the student supplies {(H, 3)} as the set representation of the event, this constraint will be violated flagging an error.

Problem	**Correct Event Set Representation**
A fair die is rolled once, then the number on the upper face is observed. **What is the probability of getting an even number?** **You can check the set representation of the event here.** **(a)**	{2, 4, 6}
	Student Event Set Representation
	{2, 3, 5} **(b)**

Constraint Parameters
Relevance condition: *if the problem statement involves even number(s)* ***Satisfaction condition:*** *the event set representation in the student answer should include the relevant even number(s) in the problem sample space* ***Word semantic description:*** *operation name* ***Problem segment description:*** *problem statement* ***Hint:*** *Please consider the even numbers* ***Axiom:*** *The even numbers are the numbers that are divisible by 2* **(c)**

Fig. 4. (a) An example MAST problem, (b) the correct event set representation and the student wrong answer, and (c) an applicable semantic constraint.

5.4 Semantic Constraints

A semantic constraint relates the student answer to the correct answer. For example, in case of a probability question, a semantic constraint ensures that the value of the probability is correct. On the other hand, in case of set representation of the event space question the semantic constraints ensure that all the elements in the student answer belong to the correct event set representation. A large number of semantic constraints ensure the event set elements, provided when solving a set representation of the event question, satisfy the operators in the problem statement. An example problem, the correct sample space, the correct set representation of the event and an applicable semantic constraint are shown in Figure 4. In the illustrated problem, a single die is tossed once and it is required to identify the probability of getting an even number. In case the students opts to solve the corresponding event set representation question, an applicable semantic constraint ensures that the shown correct event set elements are all included in the student answer. It should be noted that the question could include complicated operators, which means that more one constraint have to be checked.

<table>
<tr><th>Problem</th><th>Correct Event Set Representation</th></tr>
<tr><td rowspan="3">Two distinct dice are rolled once, then the upper faces are observed.
<b>What is the probability of the following: the sum of the appearing numbers is an even number?</b>
<b>You can check the set representation of the event here.</b>
<b>(a)</b></td><td>{(1, 1), (1, 3), (1, 5), (2, 2), (2, 4), (2, 6), (3, 1), (3, 3), (3, 5), (4, 2), (4, 4), (4, 6), (5, 1), (5, 3), (5, 5), (6, 2), (6, 4), (6, 6)}</td></tr>
<tr><td><b>Student Event Set Representation</b></td></tr>
<tr><td>{(1, 1), (1, 2), (1, 3), (1, 4)}
<b>(b)</b></td></tr>
<tr><td colspan="2"><b>Constraint Parameters</b></td></tr>
<tr><td colspan="2"><b><i>Relevance condition:</i></b> <i>if the problem statement involves the sum operator</i>
<b><i>Satisfaction condition:</i></b> <i>the sum of the number(s) between parentheses in the student answer should satisfy the condition on the operand(s) of the sum operator</i>
<b><i>Word semantic description:</i></b> <i>operations names</i>
<b><i>Problem segment description:</i></b> <i>problem statement</i>
<b><i>Hint:</i></b> Please consider the even number(s) of the sum of the elements.
<b><i>Axiom:</i></b> <i>The even numbers are the numbers that are divisible by 2 and sum means addition.</i>
<b>(c)</b></td></tr>
</table>

Fig. 5. (a) An example MAST problem, (b) the correct event set representation and student wrong answer, and (c) an applicable semantics constraint.

An example problem, the correct event set representation, and an applicable semantic constraint are shown in Figure 5. In the example, two distinct dice are rolled and their upper faces are observed. The problem requests the probability of the event in which the sum of the two numbers on the upper faces is even. In other words, there is an even condition on the sum of the two numbers. As shown in the example, the satisfaction condition considers both operations by entailing that the student answer to the set representation of the event space question should satisfy the even condition on the sum of each of the provided elements. In addition, the hints and axiom are the linguistic aggregations of hints and axioms of the two constraints since they appear as one operation for the student.

Another example is shown in Figure 6. In this example, more than one operation are found in the problem statement. Therefore, more than one constraint have to be checked to identify the satisfied and violated constraints. The validation process is based on the intermediate steps of the correct answer which are available due to employing an expert module in MAST [39]. Hints and axioms in such problems give the student the solution path that should be followed to get the correct answer. MAST has the potential to support the student with the solution path also due to utilizing the expert module component.

<table>
<tr><th>Problem</th><th>Correct Event Set Representation</th></tr>
<tr><td rowspan="3">Assume tossing a fair die once, and then tossing a single coin once in case of getting an even number. In case of appearance of a non-even number, two distinct coins are flipped once.
<b>What is the probability of the appearance of at most one Head and being a number greater than or equal 3?</b>
<b>You can check the set representation of the event here.</b>
<b>(a)</b></td><td>{(H, 3), (H, 4), (H, 5), (H, 6)}</td></tr>
<tr><td><b>Student Event Set Representation</b></td></tr>
<tr><td>{(H, 1), (H, 2), (H, 3), (H, 4),
(H, 5), (H, 6)}
<b>(b)</b></td></tr>
</table>

Constraint Parameters

Relevance condition: *if the problem statement includes the AND operator*
Satisfaction condition: *the student answer must satisfy two (conditions) of the AND operator*
Word semantic description: *violated operation*
Problem segment description: *problem statement*
Hints:

- *Consider the sample space elements that include at most one head.*
- *Consider the numbers which are greater than or equal 3.*
- *Consider the space elements that satisfy the first and the second condition.*

Axioms:

- *At least means appearing a certain number of times or less.*
- *Greater than means larger than.*
- *AND means the answer satisfies the first condition and the second condition.*

(c)

Fig. 6. (a) An example MAST problem, (b) the correct event set representation and student answer, and (c) an applicable semantics constraint and its parameters.

5.5 Sample Space Graph Constraints

The sample space question importance is related to the transformation of the question header (the story) to a mathematical form as sample space elements. Since the sample space can be too large and too time-consuming to write, as previously noted, this question is provided as a multiple-choice question in which the student can select a graph relevant to the sample space of the problem story. Sample space graph constraints address the effect of presence, sequence of the object(s), and the conditions (in advanced questions) on the sample space. The generation of the multiple choices is based on disturbing the correct graph by violating one of the constraints that should

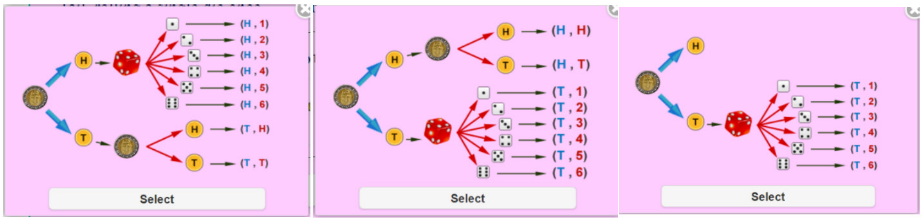


Fig. 7. Sample space question example.

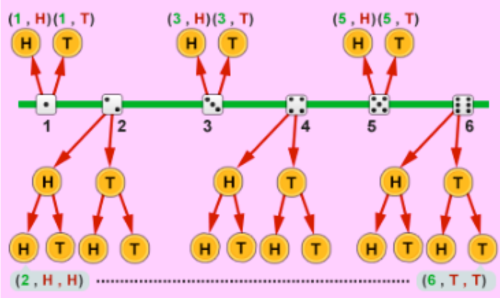
be satisfied. Accordingly, a student selection of a wrong graph is interpreted as a violation of a specific constraint.

An example of multiple choices for a sample space graph question are indicated in Figure 7, where the question presented to the student is "*A single coin is flipped once, and then a fair coin is flipped once in case of getting a Head. In case of occurrence of a Tail, a single die is rolled once. Select the correct sample space graph*". The first choice in the figure changes the condition of appearance of a coin's head to be associated by rolling a die instead of a coin and subsequently the appearance of the coin's tail is associated by flipping anther coin. The second graph in the figure is the correct one, while the third graph overlooks flipping of the second coin.

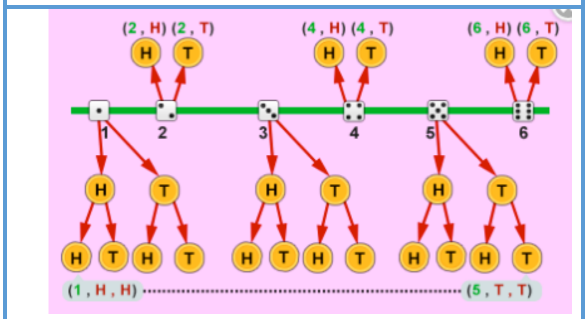
An example of a problem story, a relevant sample space graph and an applicable sample space graph constraint are shown in Figure 8. In the illustrated problem story, a single die is tossed once. If the condition of obtaining an odd number is satisfied, a fair coin is flipped once; otherwise it is flipped twice. Several constraints are applicable to such a sample space graph such as the constraint shown in Figure 8(c) which entails that the graph should correctly describe the condition in the problem header. Other sample space graph constraints ensure that the student selects the graph which considers all the objects involved in the problem story preserving their sequence. Another constraint ensures the student takes into consideration the multiplicity of the objects (in case of the existence of more than one object of the same type) or the multiplicity of an action (such as the case when a fair coin is flipped twice as in the example shown in Figure 8).

6 Constraint-based Student Modeling (CBM) in MAST

MAST employs CBM to diagnose the student errors and provide relevant help. Figure 9 shows the architecture of MAST system. MAST constraints discussed in the previous section are stored in the domain constraints database. According to the student selection of the problem type and difficulty level, or an automated selection, a problem is generated by the Problem Generation Module (PGM). The problem is presented to the student and the system prompts the student to enter his/her answer. As the student provides an answer to a problem question, the pedagogical strategy modules passes it in to the CBM module. Accordingly, the CBM module starts analysing the student answer to check its correctness and identify the violated constraints in case of an error. The analysis process starts by checking the syntax constraints and

<table>
<tr><th>Problem</th><th>Correct Sample Space</th></tr>
<tr><td rowspan="3">A single die is tossed once, and then a fair coin is flipped once in case of obtaining an odd number. In case of being an even number, a fair coin is tossed twice.
<b>(a)</b></td><td>

(1 , H)(1 , T)
(3 , H)(3 , T)
(5 , H)(5 , T)
1
2
3
4
5
6
(2 , H , H)
(6 , T , T)

</td></tr>
<tr><th>Student Sample Space</th></tr>
<tr><td>

(2 , H) (2 , T)
(4 , H) (4 , T)
(6 , H) (6 , T)
1
2
3
4
5
6
(1 , H , H)
(5 , T , T)

<b>(b)</b></td></tr>
<tr><td colspan="2"><b><i>Relevance condition:</i></b> <i>if the problem requests selecting one of the sample space graphs & the header contains a condition, which is the appearance of odd number(s)</i>
<b><i>Satisfaction condition:</i></b> <i>the student selects the graph that describes the condition</i>
<b><i>Word semantic description:</i></b> <i>operations names</i>
<b><i>Problem segment description</i></b> <i>header condition</i>
<b><i>Hint:</i></b> <i>Consider the odd numbers in the sample space.</i>
<b><i>Axiom:</i></b> <i>The odd numbers are the numbers that are not divisible by 2.</i>
<b>(c)</b></td></tr>
</table>

Fig. 8. (a) An example MAST problem story, (b) correct and student wrong answer (c), and an applicable sample space graph constraint.

then the semantic constraints are checked. It is worth mentioning that violation of some of syntax constraints indicates the student answer is fundamentally wrong and proceeds to checking the semantic constraints. For example, if the student answer for the event set representation question contains elements of an object where the question header contains another object, then the student answer is totally wrong and there is no reason to continue checking the semantic constraints.

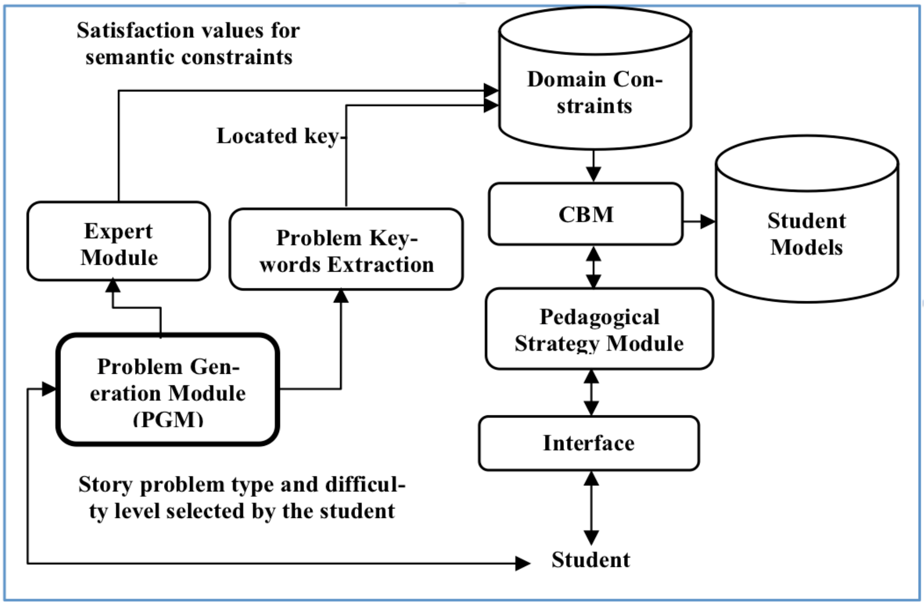

Satisfaction values for semantic constraints
Domain Constraints
Located key
Expert Module
Problem Keywords Extraction
CBM
Student Models
Pedagogical Strategy Module
Problem Generation Module (PGM)
Interface
Story problem type and difficulty level selected by the student
Student

Fig. 9. The architecture of MAST system.

The verification mechanism is based on checking the constraints relevant to the conditions presented by the problem. Only the constraints whose relevance conditions are satisfied are matched to identify whether or not their satisfaction conditions are violated. As previously noted, the relevance conditions of each constraint type are related to a specific problem segment. For example, the relevance conditions of the syntax constraints are related to the question canned text specifying the main requirements of the problem (the probability of an event or the event set representation). On the other hand, the relevance conditions of the context constraints and the sample space graph constraints are related to the problem header. Finally, the semantics constraints are related to the statement components of the question. This implies that each constraint is only matched against the parameter (keyword) extracted from its relevant problem segment for a highly efficient matching process.

The constraints whose relevance conditions match are the only constraints that fire. In other words, they are the constraints whose satisfaction conditions are matched against the student answer by the CBM module to determine whether they are satisfied or violated. For example, objects are extracted from the problem header and their count and sample spaces are extracted to be used in the matching process against the context constraints as explained earlier. In case of any error, the pedagogical strategy module is informed to provide relevant assistance to the student.

It is worth noting that the relevance and satisfaction components of all constraint types are defined and stored in the domain constraints database, except the semantic constraints whose satisfaction conditions are not predetermined. This stems from the fact that the generated questions have variable operations and associated operands, which are randomly selected. Alternatively, the expert module generates the correct answer for each problem statement and hands them in to the CBM module to be used in developing a satisfaction criterion for each semantic constraint on the fly. In case a statement includes more than one operation, all the semantic constraints that include the operations names in their relevance conditions are fired in sequence depending on the mathematical precedence of the operators. The expert module provides the result of applying each operation to be used in the matching process.

A student model database stores all the student logs regarding the selected problem types and the presented statements. The status of the student answer (correct or incorrect) in addition to the types and status (satisfied and violated) of the constraints to which the student is exposed, Trail numbers, the time spent in solving the different questions and the history of assistance provided are also recorded in the database.

7 Scaffolding in MAST

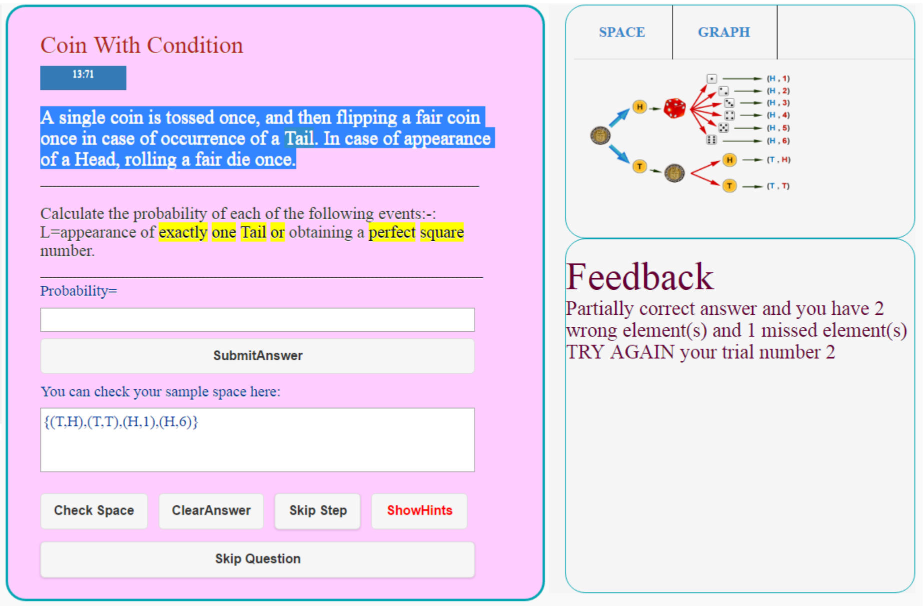
As previously noted, MAST employs two types of scaffolding techniques to assist and guide the students to provide the correct answers, namely, (i) scaffolding questions and (ii) scaffolding feedback.

7.1 Scaffolding Questions

As previously noted, in scaffolding questions, the problem solution is divided into a sequence of steps. Accordingly, the student is provided with a sequence of questions such that the answer to each question solves a problem step. In case of the probability questions of MAST, the solution steps are implicit. In other words, the student needs to implicitly derive the sample space and the correct event set representations before computing the probability, which is equal to the number of elements of the event set representation divided by the number of elements of the sample space. Accordingly, MAST provides the student with the optional scaffolding sample space graph and event set representation questions whose solution solves major steps of the problem.

Since the sample space of a given problem is common to all the included probability questions, the sample space graph scaffolding question is provided in each problem before any other probability question as shown in Figure 10. Whether the student solves this question or skips it, the sample space graph is displayed throughout the rest of the problem. The student may display the sample space elements in textual form instead of in graphical form. This implicitly solves the first step of each following probability question. Additionally, the irrelevant graphs are developed by violating some of the sample space graph constraints to detect the student error in case one of them is selected.

After deriving the correct sample space of a probability problem, the next step in solving any probability question is to implicitly compute the correct event set representation. Thus, as previously noted, each probability question in MAST is accompanied by an optional event set representation scaffolding question to allow the student to write down the event set representation used in the computation and check its validity before proceeding.



Coin With Condition

13:71

A single coin is tossed once, and then flipping a fair coin once in case of occurrence of a Tail. In case of appearance of a Head, rolling a fair die once.

Calculate the probability of each of the following events:-:
L=appearance of exactly one Tail or obtaining a perfect square number.

Probability=

{(T,H),(T,T),(H,1),(H,6)}

You can check your sample space here:

Feedback

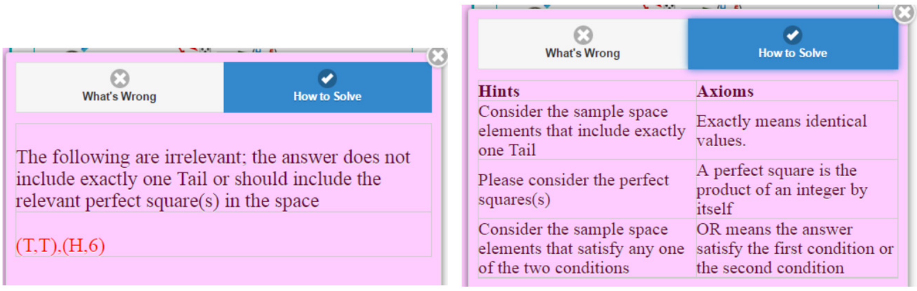
Partially correct answer and you have 2 wrong element(s) and 1 missed element(s) TRY AGAIN your trial number 2

Fig. 10. Highlighted keywords and the general feedback.

7.2 Scaffolding Feedback

MAST provides different levels of feedback. It starts by presenting a clue to the student by highlighting the related violated constraint keywords that may be misinterpreted by the student. As previously noted, MAST extends the classical representation of a constraint such that each constraint involves the word semantic description WSD and the problem segment description PSD where the corresponding keyword can be located. Thus, upon the violation of a given constraint, the related keywords are located in the problem and highlighted as a minimal form of feedback to draw the attention of the student. In addition, general feedback that indicates whether the student answer is correct or incorrect, and whether there are any missing elements or incorrect elements is also indicated. The highlighted keywords, and the general feedback, are provided automatically to the student upon submission of an answer. In other words, MAST introduces visual hints to the student in addition to the general feedback. For example, Figure 10 illustrates violation of a student answer to the shown constraint due to violation of the two highlighted operations. In addition, the general feedback shows that the student answer has some correct elements (partially correct) in addition to two wrong elements and one missed element.

In case of an error in the event set representation question, the student may request help by requesting a hint. Then the system will support the student by two types of feedback. The first is *What's Wrong*, which identifies whether there are any irrelevant elements in the answer as shown in Figure 11. The second type of feedback is *How to Solve*, which gives the hints and axioms of the operations violated in the same sequence of solving. These two types of feedback are based on the expert module out-put, which generates the correct answer and corresponding solution path. It is worth noting that MAST constraints are specified at a low-level of granularity, to allow fine-grained diagnosis of the student errors in case of compound and multiple operations. Therefore, the system is able to identify the specific cause leading to the mistakes made. The last feedback is the correct answer which is available to the student only when he/she has arrived at the last trial allowed.



The following are irrelevant; the answer does not include exactly one Tail or should include the relevant perfect square(s) in the space

(T,T),(H,6)

Hints	Axioms
Consider the sample space elements that include exactly one Tail	Exactly means identical values.
Please consider the perfect squares(s)	A perfect square is the product of an integer by itself
Consider the sample space elements that satisfy any one of the two conditions	OR means the answer satisfy the first condition or the second condition

Fig. 11. What's Wrong and How to Solve feedbacks.

8 Evaluation

The goal of the evaluation is to assess the effectiveness of MAST in tutoring the students using various tests. Moreover, we are interested in the assessment of the effect of highlighting keywords in case of errors on learning outcomes.

8.1 Method and Setup

The subjects of the test were students enrolled in the third secondary level in the Egyptian educational system. They had basic prior knowledge of probability theory, and were expected to address more complex problems using MAST as part of their curriculum. The study included 51 volunteer students who previously attended three one-hour classes about the basics of the probability theory addressed by MAST. A paper-based pre-test exam was conducted including one question from each of MAST 23 problem types. The students were then randomly divided into three groups each including 17 students as follows:

- Group G1 was allowed to practice using the textbook that included working examples, but had no access to MAST.
- Group G2 was allowed to practice using MAST, without the feature of highlighting keywords.
- Group G3 was allowed to practice using a full version of MAST with all features included.

The students were allowed to practice for two weeks after which they sat for a post-test exam. This exam included the same number and types of questions as the pre-test exam, but with different order and specifications. We conducted four types of tests as follows:

- *Paired samples t-test* was used to compare the pre-test and post-test exams of the students in each group.
- For each group, we also computed the *normalized knowledge gain* which is given by the following formula [40]:

  Normalized Knowledge Gain = (post-score - pre-score)/(100 – pre-score)

  Where *post-score* and *pre-score* are the average percentage scores of the post-test exam and the pre-test exam respectively.

- The *Mann-Whitney U test* was used for detailed comparison of the gain of the exams of the students in each two group pairs.
- *Power curves* for the constraint violations were used to evaluate the 39 different constraints. We computed for each student the percentage of violated constraints the first time each constraint was valid. We then computed the average percentage across all the students. We repeated this in subsequent events of constraints validity and drew corresponding power curves.

8.2 Findings and Discussion

The scores of the pre and post-test exams of the three groups are shown in Table 1. It is worth mentioning that the grades are out of 23 and the gain is calculated as the difference between the pre and post-test exam scores.

Using the *paired samples t-test*, we obtained t-values of 8.98, 9.32, and 9.67 for groups G1, G2 and G3 respectively. Using those t-values and degrees of freedom of 16, we obtained p-values < 0.00001 for all three groups. This indicates that the results were significant at $p \leq 0.05$. In other words, in each of the three cases, the students' performance was enhanced in the post-test in comparison to the pre-test. However, groups G2 and G3 showed higher improvement in comparison to G1 in terms of gain.

Similarly, the *normalized knowledge gain* was found to be 0.422, 0.716 and 0.768 respectively, which is significant in each of the three groups. Nevertheless, in the case of each of groups G2 and G3, it was significantly higher than in the case of group G1. It is worth noting that in the two above tests, group G3 demonstrated the highest improvement among the three groups.

Using *Mann-Whitney U test* on the different group pairs, we obtained a U-value of 64, a critical value of U of 87, and a p-value of 0.005 when comparing groups G1 and G2; a U-value of 53, a critical value of U of 87, and a p-value of 0.0014 when comparing groups G1 and G3; and a U-value of 128, a critical value of U of 87, and a p-value of 0.56 when comparing groups G2 and G3. In other words, the results in the first two cases were significant at $p \leq 0.05$, but not in the third case. This suggests that while MAST showed a positive effect on improving the grades of the students in comparison to the text book-based traditional approach, the two variants of MAST didn’t show any significant difference.

Table 1. Pre-test and post-test scores for individuals in the three groups

	Group G1			**Group G2**			**Group 3**		
	Pre-test	***Post-test***	***Gain***	***Pre-test***	***Post-test***	***Gain***	***Pre-test***	***Post-test***	***Gain***
1	18	22	**4**	11	22	**11**	12	22	**10**
2	17	23	**6**	12	18	**6**	11	22	**11**
3	16	20	**4**	13	19	**6**	14	23	**9**
4	12	18	**6**	16	23	**7**	18	20	**2**
5	11	17	**6**	17	22	**5**	16	20	**4**
6	16	18	**2**	19	21	**2**	14	19	**5**
7	19	20	**1**	18	20	**2**	15	20	**5**
8	14	16	**2**	14	21	**7**	17	21	**4**
9	13	17	**4**	15	22	**7**	12	20	**8**
10	12	14	**2**	18	21	**3**	17	22	**5**
11	11	16	**5**	17	20	**3**	15	23	**8**
12	16	18	**2**	14	22	**8**	12	19	**7**
13	15	18	**3**	19	22	**3**	18	20	**2**
14	14	17	**3**	12	20	**8**	15	22	**7**
15	17	19	**2**	11	19	**8**	14	23	**9**
16	18	20	**2**	15	20	**5**	17	22	**5**
17	17	20	**3**	16	21	**5**	16	21	**5**
Average	15.06	18.41	**3.35**	15.12	20.76	**5.65**	14.88	21.12	**6.24**
SD	±2.56	±2.26	**±1.62**	±2.69	±1.35	**±2.50**	±2.21	±1.36	**±2.66**

Figures 12 through 15 show the power curves and the log-log plots of the two groups G2 and G3 respectively. A comparison is provided in Figure 16. As shown in the figures, both versions of MAST have a positive effect of tutoring the students and reducing the average number of constraint violations. Nevertheless, MAST with the keyword highlighting feature has a slightly higher positive effect on speeding up learning since it tutors the students to recognize keywords in order to correct their errors and provide correct solutions.

To sum up, MAST had positive effects on improving the pre and post-test scores and showed considerable student gain (at least 9%). Additionally, MAST had a higher positive effect in comparison to the text-book based approach especially in the normalized knowledge gain. This may be due to the various types of assistance provided to the students in the course of learning. On the other hand, the two versions of MAST, with and without keywords highlighting in case of error respectively, did not show significant differences in this respect. However, the power curves showed speedup in learning using highlighting. This suggests that using various forms of assistance can help speedup learning and that well-designed tutoring systems providing relevant help can be superior to the text-book based traditional approach.

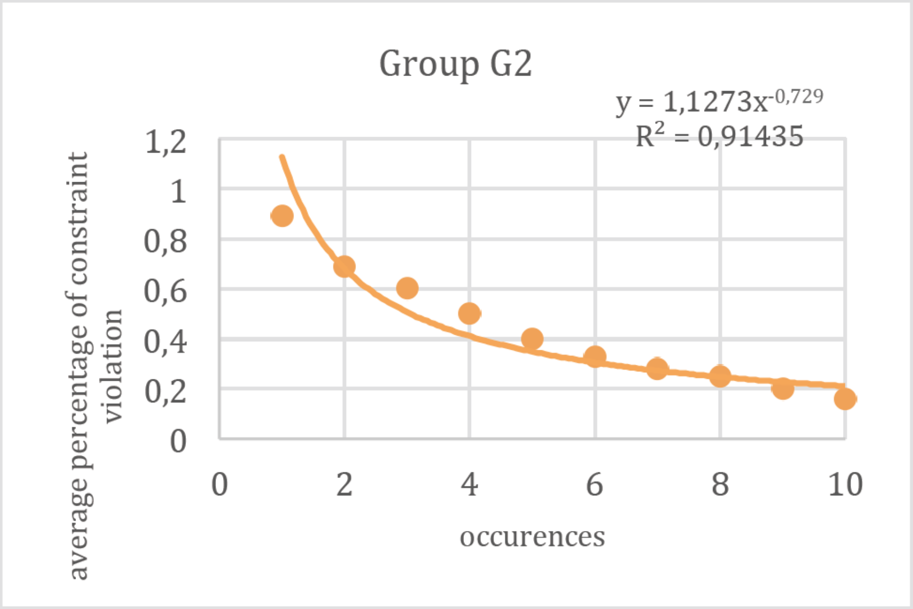


Fig. 12. The power curve of groups G2.

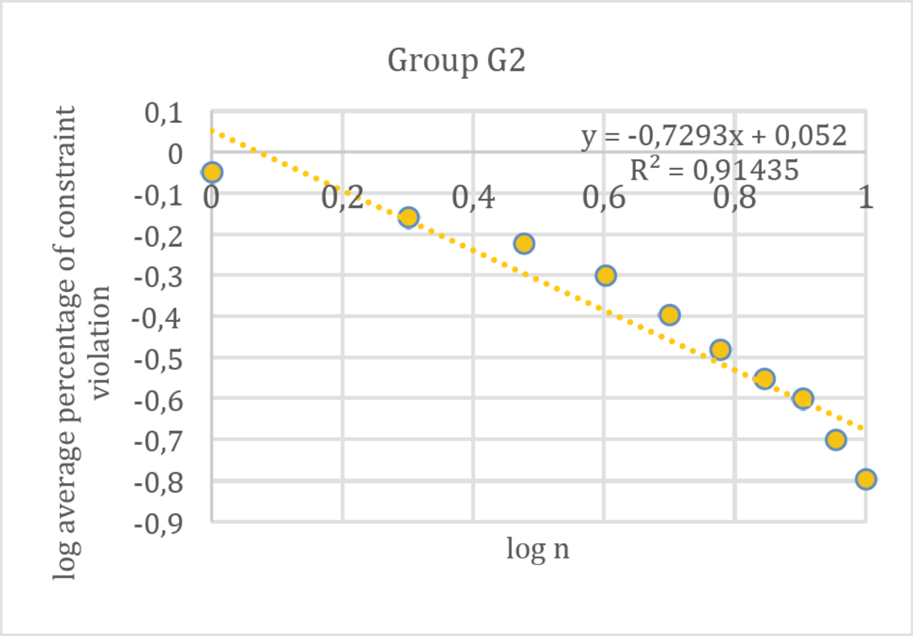


Fig. 13. The log-log plot of groups G2.

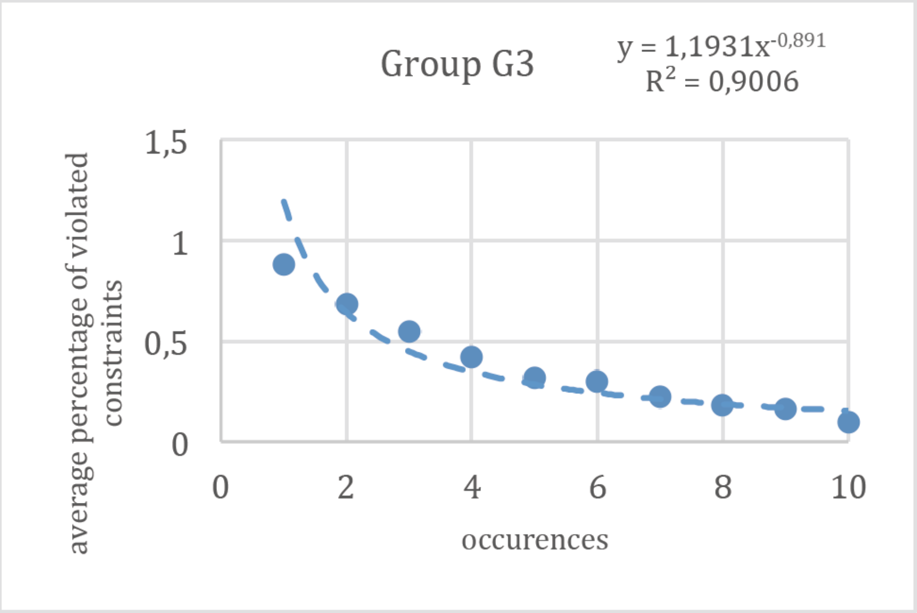


Fig. 14. The power curve of groups G3.

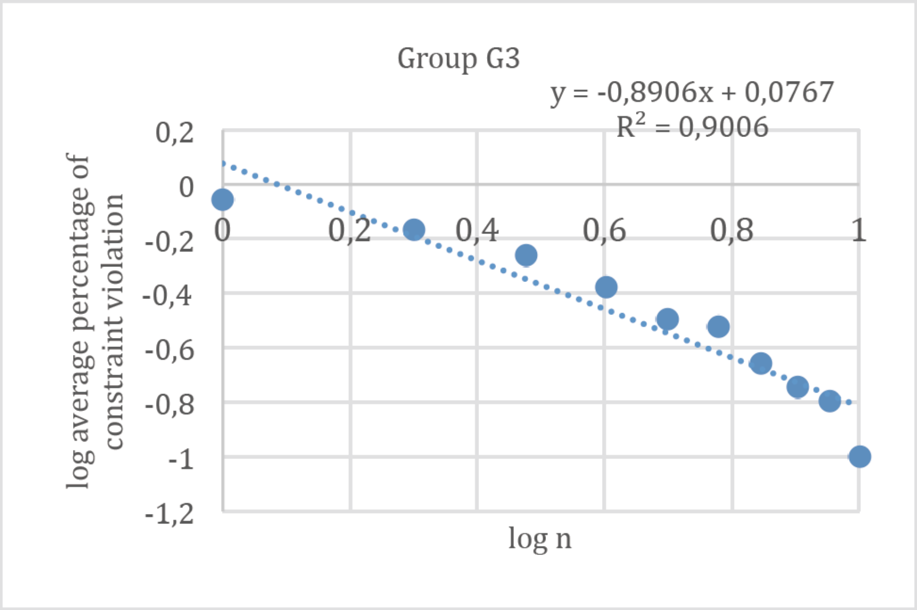


Fig. 15. The log-log plot of group G3.

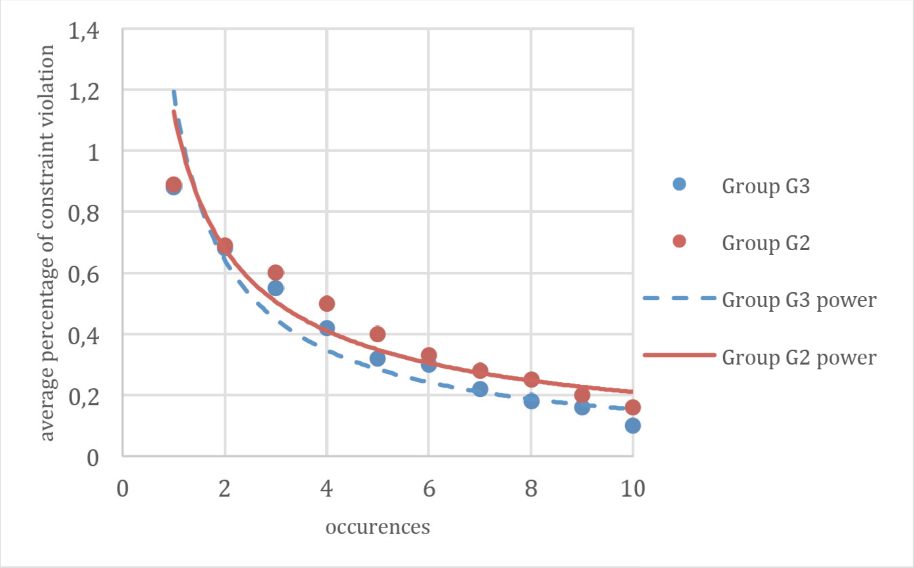


Fig. 16. Comparison of the power curves of groups G2 and G3.

9 Conclusion

This paper presented CBM in the context of MAST, a Web-based intelligent tutoring system of probability story problems. MAST has the ability to generate problems of different contexts, types and difficulty levels for self-paced learning. This paper is mainly concerned with short-term constraint-based student modeling in MAST for the goal of diagnosing the student error to provide relevant help for effective tutoring. Generation of problems in MAST is conducted using natural language techniques and hence the role and syntax of each problem segment is known. This allows MAST to relate each student error to specific keywords within a problem segment. Thus, it takes into consideration the possibility of the student error due to misinterpretation of the story problem. Accordingly, this paper extends the classical formal definition of a constraint to be able to provide various forms of help such as highlighting keywords and providing hints and axioms. The paper also introduces new types of constraints suitable for its problems. MAST constraints are specified at a low-level of granularity to allow fine-grained diagnosis of the student error and hence more specific feedback. MAST integrates CBM with scaffolding questions and feedback so that the student is guided step by step towards the correct answer. In addition to its advantage in tutoring the student, scaffolding also makes MAST able to diagnose compound errors. The different forms of scaffolding feedback provided to the students include:

- Keywords corresponding to a constraint are highlighted to draw the attention of the student to take them into consideration.
- General feedback indicates if the student answer is wrong or correct.

- Hints corresponding to the constraint are provided upon request.
- Concept axioms behind the student error(s) may also be provided to the student upon request.
- The student is also allowed to check the event set elements and in case of an error, the student may request help about what is wrong with the answer.

A preliminary empirical evaluation of MAST has been conducted, and the results indicate the potential of effectiveness of its student modeling approach in tutoring the students. The number of constraint violation decreases along the learning curve and there is an improvement in the results of the post-test exam in comparison to the pre-test exam. The novel feature of highlighted keywords identified by their roles in the generated problems show a speeding up effect. More detailed evaluation on a larger audience would further enhance this conclusion.

The system focuses on probability story problems. As future work, it can be generalized to deal with different story problems in math courses. Machine learning techniques are required to assess and model the student knowledge. Long-term student model can be utilized to demonstrate the problems in an adaptive individualized manner. Hypermedia technology can also be used to annotate the problems to allow or unlock certain problem types, depending on the student's knowledge. MAST is characterized by variability of its questions and scaffolding questions in addition to the scaffolding help to be presented.

10 References

[1] Brusilovsky, P., and Millan, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W. (Ed.), The Adaptive Web, Springer-Verlag. https://doi.org/10.1007/978-3-540-72079-9_1

[2] Merceron, A., and Yacef, K. (2005). TADA-ED for educational data mining. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 7(1): 1-13.

[3] Pahl, C., and Kenny, C. (2009). Interactive correction and recommendation for computer language learning and training. IEEE Transaction on Knowledge and Data Engineering, 21(6): 854–866. https://doi.org/10.1109/TKDE.2008.144

[4] Smith, A., and Blandford, A. (2003). MLTutor: An application of machine learning algorithms for an adaptive web-based information system. International Journal of Artificial Intelligence in education, 13: 235–261.

[5] Farzan, R., and Brusilovsky, P. (2005). Social navigation support through annotation-based group modeling. 10th International Conference on User Modeling, Edinburgh, Scotland, UK, pp. 463–472. https://doi.org/10.1007/11527886_64

[6] Roll, R., Baker, V., and Koedinger, K. (2004). What goals do students have when choosing the actions they perform? 6th International Conference on Cognitive Modeling, Pittsburgh, PA, USA, pp. 380-381.

[7] Mhlenbrock, M., and Scheue, O. (2005). Using action analysis in active math to estimate student motivation. 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems, Saarbrcken, Germany, pp. 56-60.

[8] Stash, N., and Bra, P. (2004). Incorporating cognitive styles in aha! (the adaptive hypermedia architecture). IASTED International Conference on Web-based Education, Innsbruck, Austria, pp. 378-383.

[9] Kelly, D., and Tangeney, B. (2005). 'First aid for you': Getting to know your learning style using machine learning. 5th IEEE International Conference on Advanced Learning Technologies, Kaohsiung, Taiwan. https://doi.org/10.1109/ICALT.2005.1

[10] Carmona, C., Castillo, G., and Millan, E. (2008). Designing a dynamic network for modeling students' learning style. 8th IEEE International Conference on Advanced Learning Technologies, Santander, Cantabria, pp. 346-350.

[11] Ohlsson, S., and Rees, E. (1991). The function of conceptual understanding in the learning of arithmetic procedures. Cognition & Instruction, 8(2): 103-179. https://doi.org/10.1207/s1532690xci0802_1

[12] Ohlsson, S., and Rees, E. (1991). Adaptive search through constraint violations. Journal of Experimental and Theoretical Artificial Intelligence, 3: 33-42. https://doi.org/10.1080/09528139108915280

[13] Ohlsson, S. (1992). Constraint-based student modeling. Artificial Intelligence in Education, 3(4): 429-447.

[14] Ohlsson, S. (2016). Constraint-based modeling: From cognitive theory to computer tutoring – and back again, Artificial Intelligence in Education 26: 457–473 https://doi.org/10.1007/s40593-015-0075-7

[15] Mitrovic, A. (2003). An intelligent SQL tutor on the Web. International Journal of Artificial Intelligence in Education, 13: 173-197.

[16] Suraweera, P., and Mitrovic, A. (2002). KERMIT: A constraint-based tutor for database modeling. 6th International Conference on Intelligent Tutoring Systems, Biarritz, France, pp. 377-387. https://doi.org/10.1007/3-540-47987-2_41

[17] Suraweera, P., and Mitrovic, A. (2004). An intelligent tutoring system for entity relationship modeling. International Journal of Artificial Intelligence in Education, 14(3): 375-417.

[18] Mitrovic, A., and Ohlsson, S. (2016). Implementing CBM: SQL-Tutor after fifteen years. Artificial Intelligence in Education. Artificial Intelligence in Education, 26(1):150-159. https://doi.org/10.1007/s40593-015-0049-9

[19] Mitrovic, A. (2002). NORMIT: a Web-enabled tutor for database normalization. International Conference on Computers in Education, Auckland, New Zealand. https://doi.org/10.1109/CIE.2002.1186210

[20] Mitrovic, A. (2005). The effect of explaining on learning: A case study with a data normalization tutor. Conference on Artificial Intelligence in Education, Amsterdam, Netherlands, pp. 499-506.

[21] Baghaei, N., Mitrovic, A., and Irwin, W. (2005). A constraint-based tutor for learning object-oriented analysis and design using UML. 13th International Conference on Computers in Education, Singapore, pp. 11-18.

[22] Baghaei, N., Mitrovic, A., and Irwin, W. (2006). Problem-solving support in a constraint-based tutor for UML class diagrams. Technology, Instruction, Cognition and Learning, 4(2):113-137.

[23] Mitrovic, A. (1998). Learning SQL with a computerized tutor. 29th SIGCSE Technical Symposium on Computer Science Education, Atlanta, GA, USA, pp. 307-311. https://doi.org/10.1145/273133.274318

[24] Matsuda, N., Lee, A., Cohen, W., and Koedinger, K. (2009). A computational model of how learner errors arise from weak prior knowledge. Annual Conference of the Cognitive Science Society, Austin, TX, USA, pp. 1288-1293.

[25] Chrysafiadi, K., and Virvou, M. (2013). Student modeling approaches: A literature review for the last decade. Expert Systems with Applications, 40: 4715-4729. https://doi.org/10.1016/j.eswa.2013.02.007

[26] Wang, T., and Mitrovic, A. (2002). Using neural networks to predict student's performance. International Conference on Computers in Education, Auckland, New Zealand. https://doi.org/10.1109/CIE.2002.1186127

[27] Anderson, J., Boyle, C., Corbett, A., and Lewis, M. (1990). Cognitive modeling and intelligent tutoring. Artificial Intelligence, 42(1): 7-49. https://doi.org/10.1016/0004-3702(90)90093-F

[28] Mitrovic, A., Koedinger, K., and Martin, B. (2003). A comparative analysis of cognitive tutoring and constraint-based modeling. 9th International Conference on User Modeling, Johnstown, PA, USA, pp. 313-322. https://doi.org/10.1007/3-540-44963-9_42

[29] Sawyer, R. (2006). The Cambridge Handbook of the Learning Sciences. New York: Cambridge University Press.

[30] Elazhary, H., and Khodeir, N. (2017). A cognitive tutor of Arabic word root extraction using artificial word generation, scaffolding and self-explanation. International Journal of Emerging Technologies in Learning, 12(5):36-49. https://doi.org/10.3991/ijet.v12i05.6651

[31] Heffernan, N., and Heffernan, C. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education, 24(4): 470–497. https://doi.org/10.1007/s40593-014-0024-x

[32] Feng, M., Heffernan, N., and Koedinger, K. (2010). Student modeling in an intelligent tutoring system. In Stankov, S. (Ed.), Intelligent Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation, IGI Global. https://doi.org/10.4018/978-1-61692-008-1.ch011

[33] Heffernan, N. (2001). Intelligent tutoring systems have forgotten the tutor: Adding a cognitive model of human tutors dissertation. Ph.D. thesis, Computer Science Department, School of Computer Science, Carnegie Mellon University.

[34] Xiong, X., Zhao, S., Inwegen, E., and Beck, J. (2016). Going deeper with deep knowledge tracing. 9th International Conference on Educational Data Mining, pp. 545–550.

[35] Harrington, R., and Holroyd, A. (2007). Impact of scaffolding in an intelligent tutoring system. Project report, Worcester Polytechnic Institute.

[36] Finn, B., and Metcalfe, J. (2010). Scaffolding feedback to maximize long-term error correction. Memory & Cognition, 38(7): 951-961. https://doi.org/10.3758/MC.38.7.951

[37] McArthur, D., Stasz, C., and Zmuidzinas, M. (1990). Tutoring techniques in algebra. Cognition and Instruction, 7:197-244. https://doi.org/10.1207/s1532690xci0703_2

[38] Mitrovic, A. (2010). Modeling domains and students with constraint-based modeling. In Nkambou, R. (Ed.), Advances in Intelligent Tutoring Systems, Springer-Verlag, pp. 63-80. https://doi.org/10.1007/978-3-642-14363-2_4

[39] Khodeir, N., Elazhary, H., and Wanas, N. (2017). Rule-based cognitive modeling and model tracing in a math story problem tutor. International Journal of Emerging Technologies in learning, 12(4):111-125.

[40] Hake, R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. American Journal of Physics, 66(1):64-74. https://doi.org/10.1119/1.18809

11 Authors

Nabila Khodeir is a researcher in the Informatics department at the Electronics Research Institute, Cairo, Egypt. Her research interests include intelligent tutoring systems, automatic question generation, user modeling and natural language processing. She earned her Ph.D. and ME from the Electronics and Communications department at Cairo University.

Nayer Wanas is an associate professor of informatics at the Electronics Research Institute, Giza, Egypt. His research interests include data mining and learning systems. He has also worked as a researcher at Microsoft Research on mining social data. Dr. Wanas received his Ph.D. in Systems Design Engineering from the University of Waterloo in 2003, and his Bachelors and ME from the Electronics and Communications department at Cairo University in 1992 and 1996 respectively. He has over 70 technical publications in top journals and conferences and several book chapters and patents related to his research and applied work. He has served as a reviewer of several prestigious journals and on several conference program committees and industrial panels. He is a Senior IEEE member and ACM member.

Hanan Elazhary earned her B.Sc. and M.Sc. degrees from the Department of Electronics and Communications Engineering, Cairo University. She earned her Ph.D. degree in Computer Science and Engineering from the University of Connecticut, USA. Currently, she is an associate professor in the Computer Science Department, Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia and the Computers and Systems Department, Electronics Research Institute, Cairo, Egypt. Here research interests include distributed systems, software engineering and intelligent tutoring systems.

Article submitted 08 July 2017. Published as resubmitted by the authors 15 September 2017.