Mathematics Intelligent Learning Environment

Hongguang Fu [1], *Xiuqin Zhong* [1,*], *Zhong Liu* [2]

zhongxiuqin2009@gmail.com

[1] University of Electronic Science and Technology of China, 611731, Chengdu, China
[2] Leshan Vocational & Technical College, 614000, Leshan, Sichuan, China

Abstract: *An interactive intelligent learning system in mathematics is the trend of educational technology, where students can practice homework problems and take practice tests online. However, developing such a desirable online environment is still an open research area and far from perfect. In this paper, we will present a Mathematics Intelligent Learning Environment (MILE) that provides capability for automatic theorem proving for geometry and automatic equations solving for algebra. Initial design is focused to provide an interactive intelligent learning environment for junior high schools mathematics www.ihomework.com.cn. MILE automatically checks if a student's homework is correct step by step. The system not only determines the correctness for each step, but also provides assistance on each step if a student chooses option for help. It is therefore a powerful learning tool for students that acts working as a personal tutor.*

1. Introduction

In recent years, the existence of an interactive intelligent learning environment in mathematics has become increasingly important for educational technology and online learning, where students can practice homework problems and take practice tests anywhere and anytime. Interest in this area has grown significantly in the last two decades, stimulated by numerous and varied studies and research work done on mathematics for primary school students or elementary students.

The term intelligent learning environment (ILE) refers to a category of educational software in which the learner is put into a problem solving situation. A learning environment is quite different from traditional courseware based on a sequence of questions, answers and feedback. The best known example of a learning environment is a flight simulator: the learner does not answer questions about how to pilot an aircraft, but learns how to behave like a "real" pilot in a rich flying context. In summary, we use the word intelligent learning environment for learning environments which include a problem solving situation and one or more agents that assist the learner in their task and monitor their learning.

Brusilovsky et al [1] defines ILEs as a combination of an intelligent tutor system (ITS) (that responds to individual students' actions and needs through the use of a student model) and a learning environment that allows for student-driven learning (e.g.: through the use of an open learner model where students' can view and customize their student model and learning process). Wenger [2] points out three types of knowledge important to intelligent tutoring, and by extension also crucial to an effective ILE: (1) knowledge about domain, (2) knowledge about tutoring, (3) knowledge about the student and student model.

Aris and Nazeer [3] proposes a web-based model of mathematics with the feature to guide the user step by step incorporated in the proposed model. MATHia and Cognitive Tutor software implemented by Carnegie Mellon University, see Carnegie Learning [4], provides step-by-step instruction and individualized support for all students in mastering mathematic skills and processes, which is based on Adaptive Control of Thought, see Anderson, Matessa and Lebiere [5]. But they are developed only for special models, so the expansion is restricted.

Fournier-Viger et al [6] proposes a method of extracting patterns from user solutions to problem-solving exercises and automatic learning task model. In addition, it can extract temporary patterns from a tutoring agent's own behavior when interacting with learner(s). Chi and VanLehn [7] provides an interactive tutoring system with teaching a domain-independent problem-solving strategy, which includes backward chaining (i.e. solving problems from goals to givens) and principle-emphasis skills (i.e. drawing students' attention to the characteristics of each individual domain principle). These systems provide intelligent learning environments with problem solving processes; however, they only focus on learning but not practicing.

Norzaidah et al [8] provides a systematic view of implementing two different artificial intelligence techniques which are rule based and case based reasoning in an intelligent tutoring system for primary school children in the subject of Mathematics. But it can not execute automated checking.

Okita [9] examines self-assessment for learning through the application of creative computer tools that can help students assess and correct their own learning, but the literature concludes that students are not usually inclined to check their own answers. Students find it relatively motivating to catch other people's mistakes. We note the method mentioned in Okita [9] is manipulated manually.

Therefore, developing such a desirable online environment is still an open area and far from perfect, mainly due to the lack of intelligence needed to support automated reasoning or automated checking.

In this paper, we will firstly describe the automated reasoning and automated checking respectively, and then set up a mathematics interactive intelligent environment based on these theories. Initial design is focused on providing an interactive learning environment for junior high school mathematics. Finally, we carry out some experiments and test the environment.

2. Automated Reasoning Engine

MILE provides automatic theorem proving for geometry and automatic equation solving for algebra. It produces traditional readable proofs automatically for a geometry problem, or provides the readable solution process for an algebraic problem. The readable process will be helpful for mathematics education in the area of pedagogy.

2.1 Geometry Prover

2.1.1 Input of Geometry

There are two input modes of geometry which will be illustrated in details below.

The first input mode is constructing a dynamic geometry graph that satisfies geometric constraints by selecting objects such as point, line, circle or others, and selecting relationships such as parallel, perpendicular, angle bisector or others using Dynamic Geometry System (Math XP) Fu and Zeng [10].

Then the system will automatically generate the corresponding conditions and conclusions according to the graph. We use the following Example 1 for demonstration.

Example 1. Let ABCD be a random quadrangle, and points E, F, G, H be the mid- points of segments AB, BC, CD and DA respectively. To provide the input, MILE responds: Proof: quadrangle EFGH is a parallelogram. The dynamic geometry graph with Known and Conclusion is shown in Figure 1.

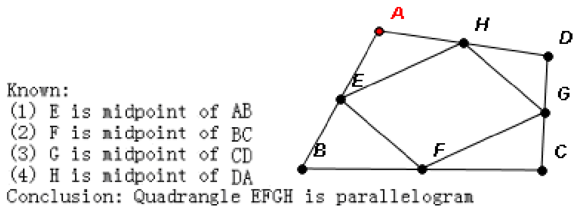
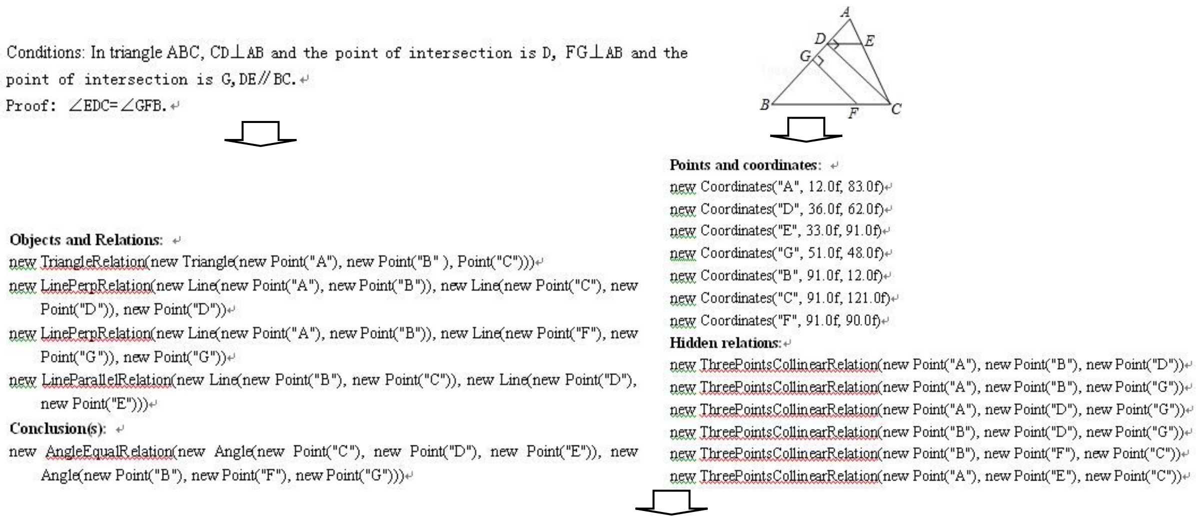


Figure 1: Input of Dynamic Geometry Graph

The second input mode is natural language texts with their respective images. On one hand, MILE can recognize objects and relations from the texts by natural language processing tools; on the other hand, MILE can extract points, coordinates of points and hidden relations from images by image processing tools. In addition, MILE converts all objects and relations into the first-order predicate logic form by integrating the texts semantics with the image semantics, and then obtain formal conditions and conclusions. We remark that currently the conditions and conclusions need to be converted into Lisp form due to the fact that at the moment the automated reasoning engine is implemented using the Lisp program language. We use the following Example 2 to demonstrate this second input mode.

Example 2: We are given the conditions: In triangle ABC, CD⊥AB and the point of intersection is D, FG⊥AB and the point of intersection is G, DE∥BC. We need to prove: ∠EDC=∠GFB. The whole process can be illustrated in Figure 2.



```
(;;;conditions
    ((1 0) (TRIANGLE A B C))
    ((2 0) (PT (LINE C D) (LINE A B)))
    ((3 0) (IP D (LINE C D) (LINE A B)))
    ((4 0) (PT (LINE F G) (LINE A B)))
    ((5 0) (IP G (LINE F G) (LINE A B)))
    ((6 0) (// (LINE D E) (LINE B C)))
    ((7 0) (COL (POINT A) (POINT B) (POINT D)))
    ((8 0) (COL (POINT A) (POINT B) (POINT G)))
    ((9 0) (COL (POINT A) (POINT D) (POINT G)))
    ((10 0) (COL (POINT B) (POINT D) (POINT G)))
    ((11 0) (COL (POINT B) (POINT F) (POINT C)))
    ((12 0) (COL (POINT A) (POINT E) (POINT C))))
(;;;conclusion(s)
    (= (:ANGLE C D E) (:ANGLE B F G)) )
(;;;points and coordinates
    (A 12.0 83.0)   (D 36.0 62.0)   (E 33.0 91.0)   (G 51.0 48.0)
    (B 91.0 12.0)   (C 91.0 121.0)   (F 91.0 90.0))
(;;;segments
    (A B) (B C) (C A) (C D) (D A) (D B) (F G)
    (G A) (G B) (D E) (D G) (A E) (E C) (B F) (C F)) )
```

Figure 2: Input of Natural Language Texts and its Images

By the way, the conclusion still holds if the point D does not lie within the line segment joining A and B.

2.1.2 Rules selecting

In the MILE system, there are 187 geometry axioms, definitions and theorems and 27 basic rules of algebra being specified from elementary geometry textbooks. We call these rules, which are all collected in “rule” table as is shown in Figure 3. The rule table contains Use, Index, Name, Property and Content. These rules are basis to perform automated reasoning.

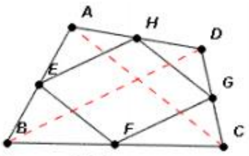
Use	Index	Name	Property	Content
●	11	Definition of midpoint	Definition	If point C divides segment AB into two equal segments, point C is the midpoint of segment AB.
●	12	Property of midpoint	Theorem	If point C is midpoint of segment AB, segment AC is equal to segment CB.
●	13	Collinear definition	Definition	If angle ACB =180 degrees, points A, C, B are collinear.
●	14	Collinear property 1	Theorem	If three points A, B, C are collinear , point C lies between points A, B, then angle ACB =180 degrees and AB =AC+BC.

Figure 3: Rule Column

Generally speaking, all rules will participate in the automated reasoning. Of course, if the user of the system doesn’t want to use some of the rules during the process of reasoning, he or she may hide those unused rules by clicking the green button of “Use” in front of the corresponding rule.

2.1.3 Automated Theorem Proving

Our current MILE system can execute automated reasoning, that are based on users’ inputs (including conditions and conclusions), and its reasoning engine which is based on the rules. Hence, MILE can generate readable proof, and can even add auxiliary lines as needed (as is shown in Figure 4). We note the readable proof is the shortest path from conditions to conclusions.




```
Known:
 (1) E is midpoint of AB
 (2) F is midpoint of BC
 (3) G is midpoint of CD
 (4) H is midpoint of DA
Conclusion: Quadrangle EFGH is parallelogram

Proof:
(5)   EH//DB     byTriangle's midplace line  from (4 1 )
(6)   FG//DB     byTriangle's midplace line  from (3 2 )
(7)   EF//CA     byTriangle's midplace line  from (2 1 )
(8)   GH//CA     byTriangle's midplace line  from (3 4 )
(9)   EF//GH     byParallel transfer   from (7 8)
(10)  EH//FG     byParallel transfer   from (5 6)
(11)  ▱ EFGH     byDefinition of parallelogram  from (9 10)
```

Figure 4: Automated Theorem Proving

If students select a different set of rules, they can obtain different problem proving processes. For example, by shutting down the rule "71 Definition of parallelogram" (as is shown in Figure 5), it can generate another proving process (as is shown in Figure 6).

●	69	Sum of interior angles of ...	Theorem	The sum of Interior angles of a polygon of n sides is (n- 2) *180 degrees.
●	70	Sum of exterior angles of...	Theorem	The sum in degrees of the exterior angles of any polygon is 360 degrees.
●	71	Definition of parallelogram	Definition	The quadrilateral with two opposite sides parallel is called a parallelogram.
●	72	Property 1 of parallelogra...	Theorem	The opposite angles of a parallelogram are equal.
●	73	Property 2 of parallelogra...	Theorem	The opposite sides of a parallelogram are parallel and equal.
●	74	Property 3 of parallelogra...	Theorem	The diagonals of a parallelogram bisect each other.
●	75	Determination 1 of parall...	Theorem	The quadrilateral in which two pairs of opposite angles are equal respectively is a parallelogram.
●	76	Determination 2 of parall...	Theorem	The quadrilateral in which two pairs of opposite sides are equal respectively is a parallelogram.

Figure 5: Shut Down Rule 71

Proof:

(5) $HE = 1 * \frac{1}{2} * DB$ byTriangle's midplace line from (4 1)

(6) $GF = 1 * \frac{1}{2} * DB$ byTriangle's midplace line from (3 2)

(7) $EF = 1 * \frac{1}{2} * CA$ byTriangle's midplace line from (2 1)

(8) $GH = 1 * \frac{1}{2} * CA$ byTriangle's midplace line from (3 4)

(9) quadrilateral EFGH Trivially

(10) EF = GH bytransfering equality from (7 8)

(11) HE = GF bytransfering equality from (5 6)

(12) parallelogram EFGH byDetermination 2 of parallelogram from (9 10 11)

Figure 6: the other Proving Process

In fact, during the reasoning process, a lot of additional relationships will be generated at the same time, which can produce a large geometry knowledge base with new produced objects and relations (as is shown in Figure 7). Consequently, we can execute the searching that is knowledge based, which is helpful for automated checking.

```
(389 ((389 388 386) (= (ANGLE B G A) (ANGLE C E A) (ANGLE C F B) 180)))
(390 ((390 4 13) (= (ANGLE B G A) 180)))
(391 ((391 4 13) (= (ANGLE B D A) 180)))
(392 ((392 391 389) (= (ANGLE B D A) (ANGLE B G A) (ANGLE C E A) (ANGLE C F B) 180)))
(393 ((393 4 13) (= (ANGLE B D A) 180)))
(394 ((394 6 13) (IP D (LINE D E) (LINE A B))))
(395 ((395 6 13) (IP G (LINE G E) (LINE A B))))
(396 ((396 6 13) (IP D (LINE D C) (LINE A B))))
(397 ((397 6 13) (IP G (LINE G C) (LINE A B))))
(398 ((398 6 13) (IP D (LINE D F) (LINE A B))))
```

Figure 7: Part of Knowledge Base

2.2 Algebra Solver

The existing symbolic computation platforms such as Maple, Mathematica, Maxima etc, are mostly designed for scientific research by using advanced mathematical methods. Typically they lack problem solving processes, or they produce unreadable problem solving process. For example, it is estimated that if we were to prove the Five Circle Theorem using the method of characteristic set, it may require millions of pages of A4 paper to show their proof process Li [11]. As a result, it is impossible to check if the proof is right or wrong by human, which we call an unreadable proof.

By simulating the cognitive models from the problem solving process of human beings, we design and implement solutions of algebraic problems by including “Simplify (trigonometric expressions)”, “Solve equation(s) in Real”, “Solve Inequality”, “Definition of Function” and etc. The input is the corresponding representation that satisfied the model (as is shown in Figure 8), and the output is readable problem solving process (as is shown in Figure 9).

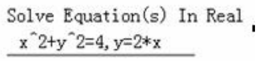


Figure 8: Example of Input Equation(s)

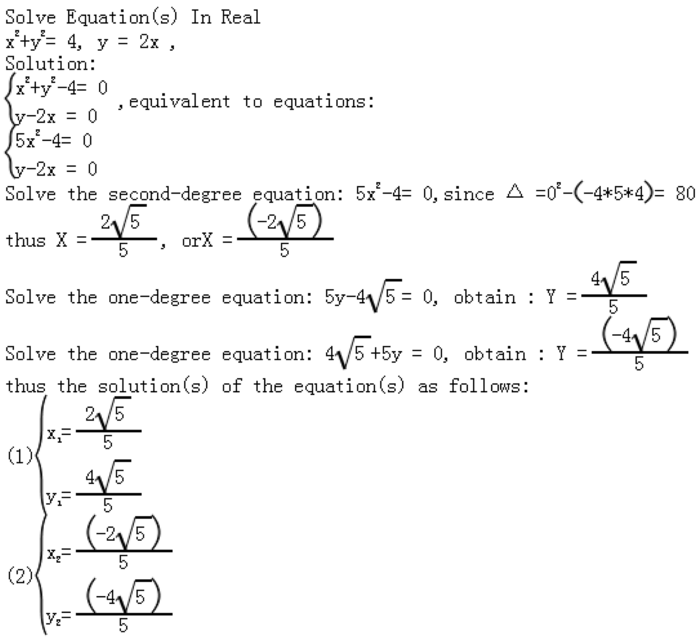


Figure 9: Readable process of Figure 8

3. Automated Checking Engine

Automated checking engine can check a student’s solution, or check whether the solving process is correct step by step. The system can not only determine the correctness for each step, but also point out the possible reasons if there are errors in student’s solutions. There are two checking models: one is objective problem (with only result) checking, and the other is subjective problem (with proving process) checking.

3.1 Objective Problem Checking

During objective problem checking, we can execute direct comparing or pattern matching if standard answer is provided. Otherwise, we will convert the objective problem to subjective problem, and obtain the needed result by automated reasoning. Finally, the system gives out the right or wrong checking result.

3.2 Subjective Problem Checking

During subjective problem checking, we need to normalize the objects and relationships firstly for the diversifications of students' solving process handwriting. Secondly, we conduct the syntax detecting. Finally, we carry out automated checking by comparing or matching with standard answer, or by automated reasoning based on geometry knowledge base and algebraic calculation of Maple, or by machine learning models from geometry knowledge base and process of standard answer, or even by numerical testing.

Consequently, the checking results such as right( ), wrong ( ), or uncheckble( ) will be displayed to the corresponding student. The automated checking result for one solution of Example 1 in Figure 1 is shown in Figure 10, and the automated checking result for one algebraic problem is shown in Figure 11.

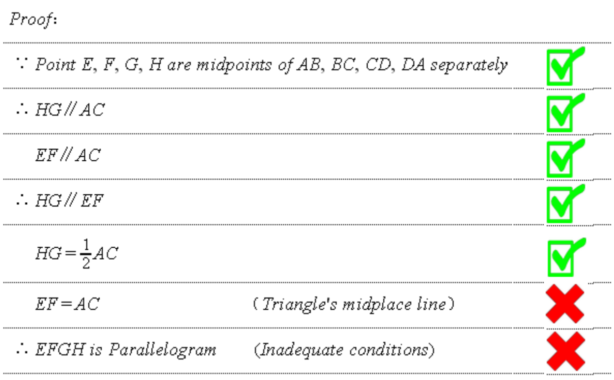


Figure 10: Automated Checking Result of Geometry

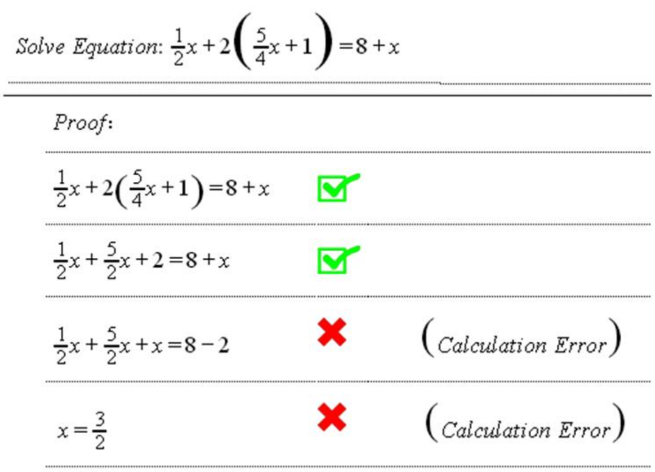


Figure 11: Automated Checking Result of Algebra

4. The Structure of MILE

A prototype of MILE for junior school-work mathematics is implemented based on an automated reasoning and automated checking system using java. Experimental environment parameters are as follows: CPU is Intel Core (tm) i3-350 Processor 3.40 GHz; Memory is 4.00 GB; Operating system is windows 7; Environment is jdk 1.7.0. The frame-structure of the environment is shown in Figure 12. MILE consists of selecting model, filling model and solving model for students. In the following, we will describe these models one by one.

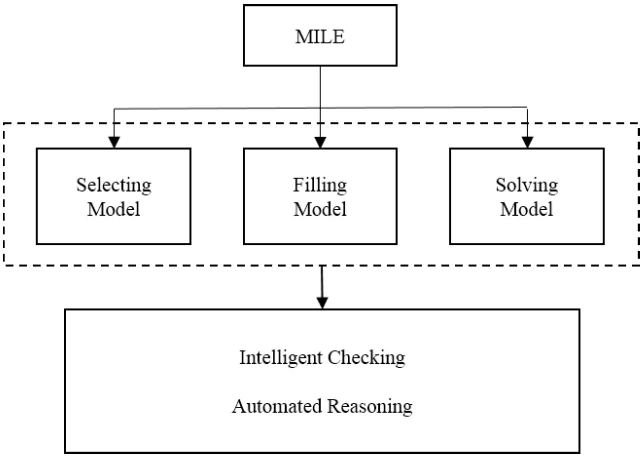


Figure 12: Framework of MILE

4.1 Selecting Model

According to the conditions and conclusions of an exercise, by simulating human being's problem solving process, the system will construct a problem solving path from conditions to conclusions based on rules, and simultaneously convert complex proof and solving process into the form of multiple-choice. Therein, more options will be listed, and students can choose the possible intermediate conclusion and select corresponding reasoning rule by clicking the items (as is shown in Figure 13). If the option is correct, the answer will be credited, otherwise the answer will be penalized.

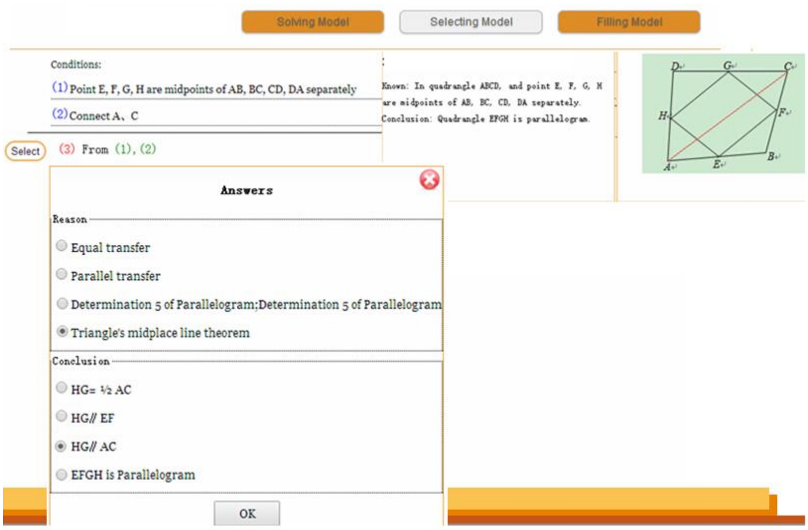


Figure 13: Selecting Model

4. 2 Filling Model

The generation process is the same as in the selecting model. All possible conclusions will be calculated. Furthermore, more options will be listed, and students can choose one of them and then drag it into the corresponding line (as is shown in Figure 14). If the option is correct, the answer will be credited, otherwise the answer will be penalized.

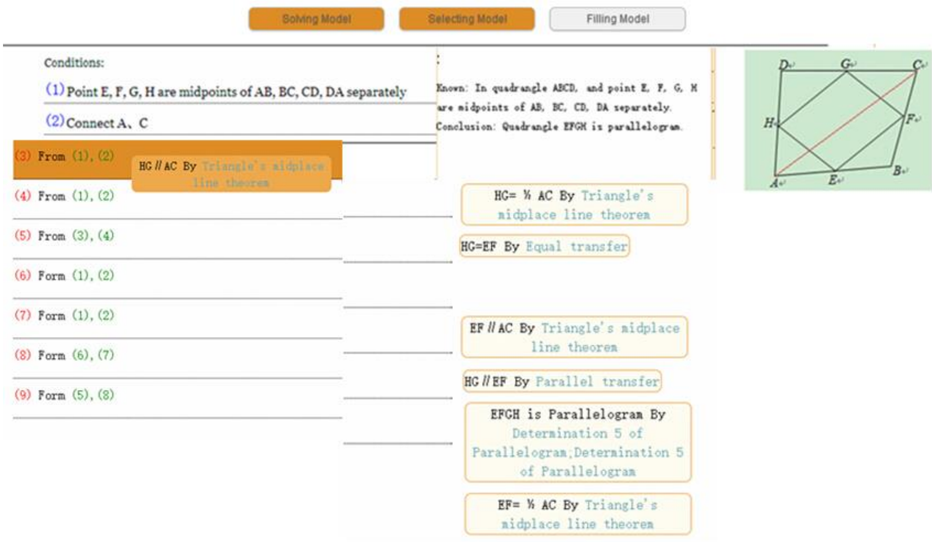


Figure 14: Filling Model

4. 3 Solving Model

In this model, students can input the problem solving process freely step by step (as is shown in Figure 15), according to their own thinking and problem-solving methods.

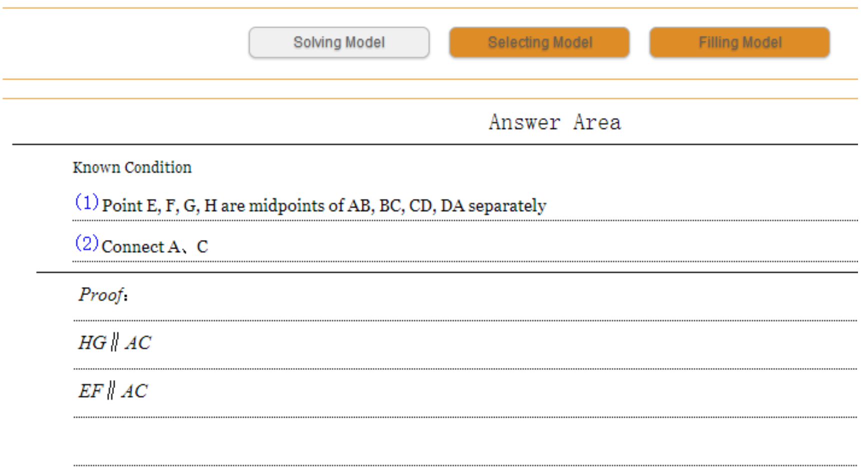


Figure 15: Solving Model

Above all, selecting model and filling model are both based on interactive and cognitive learning models, by the method of converting subjective problems into objective problems, so the automated checking both belongs to objective problem checking. Obviously, solving model checking belongs to subjective problem checking. Anyway, student's answer can be checked by our interactive intelligent checking proof.

5. Conclusion

By now, about 580 exercises of junior school mathematics are stored in our prototype of MILE, and internal α testing is executed on subjective problem checking for the exercises. Also about 100 exercises of junior school mathematics are stored in our MILE, and internal α testing on objective problem checking for these exercises. Overall, the accuracy is up to 70%, and the average time is 5 seconds for each one by statistics. By the way, the environment is currently developed for Chinese students, and only some examples are demonstrated in English. But we'll try our best to perfect the environment and develop the corresponding English version as soon as possible.

Furthermore, MILE can free a tutor from the tedious checking task, at the same time, the big data analysis can provide overall analytical data for teachers. In this way teachers can focus on finding and analyzing problems of students, and further improve instruction. ILE can determine the correctness of every step, point out the corresponding error type if possible, then discover the learning status and the problems in time, and further recommend personalized exercises adaptively.

In the future, we'll do more and deep research in automated reasoning and automated checking, and plan to develop a problem solving robot. Firstly the test papers will be scanned into the computer, and the characters and figures will be recognized by OCR. Then convert them into machine-understandable semantic forms. Later carry out symbolic computation and automated reasoning to solve problems. Finally print out results with solving process like ones given by man. Our goals in the solving problem robot can finish 80% problems in the college entrance examination in Beijing of China by 2017.

Acknowledgements

The authors would like to thank professor Wei-Chi Yang in Radford University for his valuable comments and sincere help. The work is supported by National Natural Science Foundation of China (Grant Nos. 61073099 and 61202257), the National High Technology Research and Development Program of China (863 Program) (Grant No. 2015AA015408), the Sichuan Provincial Science and Technology Department (Grant No. 2012FZ0120), UESTC education reformation and research project (Grant No. 2015XJYYB038), and China Scholarship Council.

References

[1] Brusilovsky, P., Schwarz, E., and Weber, G. (1996). ELM-ART: an intelligent tutoring system on the World Wide Web. In Intelligent Tutoring Systems, LNCS 1086, edited by C. Frasson, G. Gauthier, and A. Lesgold, pp. 261–269. Berlin: Springer

[2] Wenger, E. (1987). Artificial Intelligence and Tutoring Systems. Los Altos: Morgan Kaufmann.

[3] Teh Noranis Mohd Aris, Shahrin Azuan Nazeer. A Model of a Mathematics Editor using Interactive Agent Technology. Procedia Social and Behavioral Sciences 8(C), 588–593.2010.

[4] http://www.carnegielearning.com/

[5] J.R. Anderson, M.P. Matessa, C. Lebiere. ACT-R: A theory of higher level cognition and its relation to visual attention. Human Computer Interaction, 1997, 12(4), 439-462.

[6] Fournier-Viger, P., Faghihi, U., Nkambou, R. & Mephu Nguifo, E.. Exploiting Sequential Patterns Found in Users' Solutions and Virtual Tutor Behavior to Improve Assistance in ITS. Educational Technology & Society, 13 (1), 13–24. 2010.

[7] Chi, M. & VanLehn, K. Meta-Cognitive Strategy Instruction in Interactive Tutoring Systems: How, When, and Why. Educational Technology & Society, 13 (1), 25–39. 2010.

[8] Norzaidah Md Noh, Azlin Ahmad, Shamimi Ab. Halim and Azliza Mohd Ali. Interactive Tutoring System Using Rule-Based And Case-Based: A Comparison, IProcedia - Social and Behavioral Sciences 67, 454-463. 2012.

[9] Sandra Y. Okita. Learning from the folly of others: Learning to self-correct by monitoring the reasoning of virtual characters in a computer-supported mathematics learning environment. Computers & Education 71, 257–278. 2014.

[10] Fu Hongguang and Zeng Zhenbing. A New Dynamic Geometry Software with a Prover and a Solver, Proc. of Vienna Int. Symp. On Integral Technology into Mathematical Education (Visit-Me2002). July 2002, Vienna, Austria. ISBN 3-901769-49-8. 2002.

[11] Hongbo Li. On Miquel's Five-Circle Theorem. MM Research Preprints, 166-177, MMRC, AMSS, Academia Sinica. No. 23, December 2004.