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 users 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 an 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. The MILE can automatically check if a student's homework is correct step by step. The system not only can determine the correctness for each step, but also provide assistance on each step if student chooses option for help. It is therefore a powerful learning tool for students working as a personal tutor.*

1. Introduction

In recent years, demanding 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. Interests in this area have 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, and he 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 his task and monitor his learning.

Brusilovsky defines ILEs as a combination of an ITS (that responds to individual students' actions and needs through the use of an 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 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.

Literature [1] proposes a web-based model of mathematics with the feature to guide the user step by step is incorporated in the proposed model. Carnegie Learning [2] MATHia and Cognitive Tutor software implemented by Carnegie Mellon University 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—Rational [3]. But they are developed only for special models, so the expansion is restricted.

Literature [4] 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). Literature [5] shows that an interactive tutoring system teaching a domain-independent problem-solving strategy, which includes the backward chaining (i.e. solving problems from goals to givens) and the principle-emphasis skill (i.e. drawing students' attention to the characteristics of each individual domain principle). These systems provide intelligent learning environments with problem solving process; however, they only focus on learning but not practicing.

Literature [6] 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.

Literature [7] 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 [7] is manipulated manually.

Therefore, developing such a desirable online environment is still an open area and far from perfect, which mainly due to 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. Finally, we carry on some experiments and test the environment.

2. Automated Reasoning Engine

The system provides an automatic theorem proving for geometry and automatic equations solving for algebra. Initial design is focused on providing an interactive learning environment for junior high schools mathematics.

It will produce traditional readable proof automatically for a geometry problem or give out the readable solution process for an algebraic problem. The readable process will be helpful for the 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 satisfied 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 Tool (Math XP) . (By the way, the Dynamic Geometry Tool in mobile platform such as iphone and ipad is also implemented.) 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 are middle points of segments AB, BC, CD and DA respectively. Proof: quadrangle EFGH is a parallelogram. The dynamic geometry graph with Known and Conclusion is shown in Figure 1.

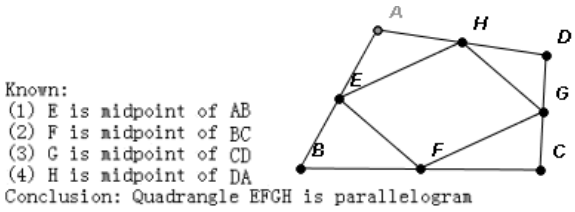


Figure 1: Input of Dynamic Geometry Graph

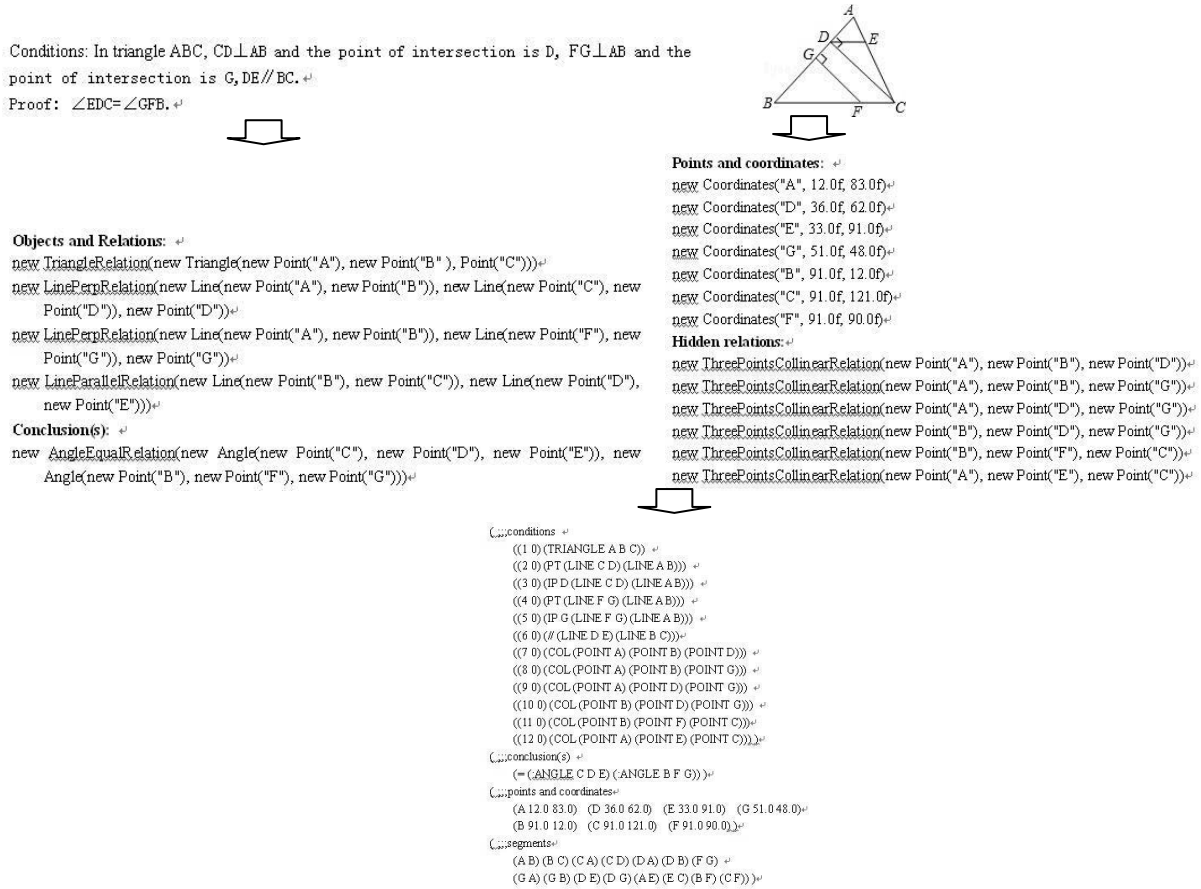


Figure 2: Input of Natural Language Texts and its Images

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 the conditions and conclusions need to be converted into Lisp form currently due to the fact that the automated reasoning engine is implemented using Lisp program language at the moment. 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.

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 to be "rules", which are all collected in "rule" column as is shown in Figure 3. The rule column contains Use, Index, Name, Property and Content. These rules are basis to perform automated reasoning.

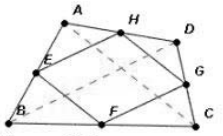
Use	Index	Name	Property	Content
	11	Definition of midpoint	Definition	If point C divides segment AB into two equal segments, point C is the midpoint of segment AB.
	12	Property of midpoint	Theorem	If point C is midpoint of segment AB, segment AC is equal to segment CB.
	13	Collinear definition	Definition	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 you don't want to use some of the rules during the process of reasoning, you 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, the MILE can generate readable proof, and even can 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 users select 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 the other 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

As for the existing symbolic computation platforms such as Maple, Mathematica, Maxima and etc, they are mostly designed for scientific research by using advanced mathematical methods. Typically they are lack of problem solving process, 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 will require millions of pages of A4 paper to show their proof process. As a result, it is impossible to check if the proof is right or wrong by human, which we call it 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).

```
Solve Equation(s) In Real
x^2+y^2=4, y=2*x
```

Figure 8: Example of Input Equation(s)

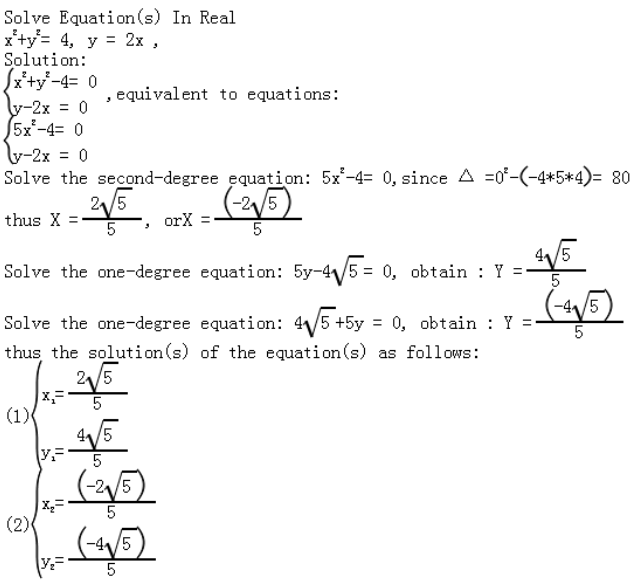


Figure 9: Readable process of Figure 8

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 on student's solutions. There are two checking models: one is objective question (with only result) checking, and the other is subjective question (with proving process) checking.

3.1 Objective Question Checking

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

3.2 Subjective Question Checking

During subjective question 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 Sympy, or by machine learning models from geometry knowledge base and process of standard answer, or by even 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.

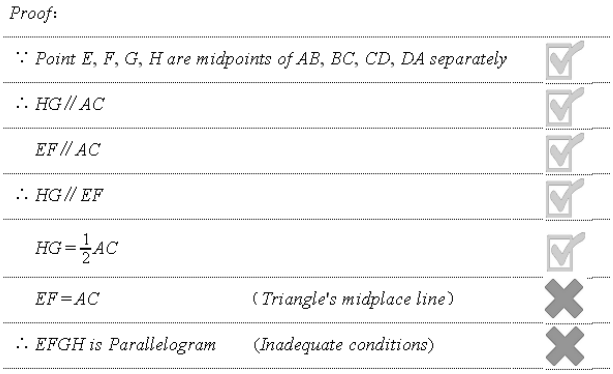


Figure 10: Automated Checking Result

4. The Structure of MILE

The MILE for junior school mathematics is implemented based on automated reasoning and automated checking. The frame-structure of the environment is shown in Figure 11. The MILE consists of Teacher's Module and Student's Module. In Teacher's Module, it mainly includes Class Management, Record Platform, Homework Management and Homework Analysis. In Student's Module, it mainly includes Homework Training (assigned by teachers), Synchronous Learning (consistent with textbook), Collaborative Learning (consistent with related students) and Free Learning. Furthermore, all functions in Student's Module can supply models such as selecting model, filling model and solving model for students, and then carry on interactive intelligent checking. In the following, we will describe some important models in detail below.

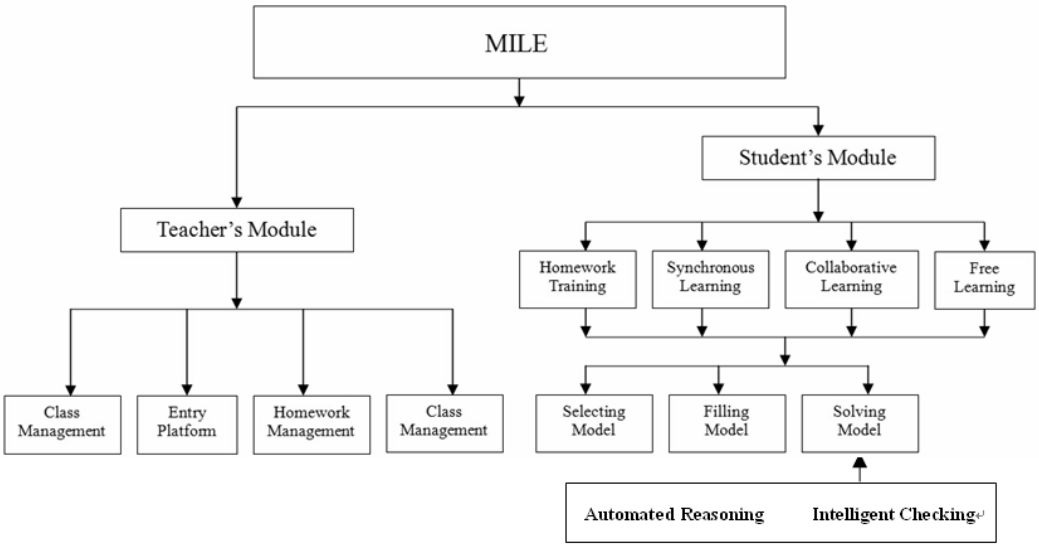


Figure 11: Framework of MILE

4.1 Teacher's Module

Texts, images, and mathematics formulas in Latex can be edited in this module. On one hand, teachers can directly edit a document of problem solving processes with step by step reasoning rules, and then import this document into the system. On the other hand, teachers can edit exercises by adding text, formula, image,

conditions, intermediate conclusions, reasoning rules from conditions to conclusions and so on. Teachers can also assign homework through this platform.

4.2 Student's Module

Intelligent Checking based on automated checking in Section 3 is included in student's module. Besides, there are three models of doing exercises in student's module, including selecting model, filling model and solving model. In the following, we'll describe them one by one.

4.2.1 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 cause and effect logic of 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 12). If the option is correct, the answer will be credited, otherwise the answer will be penalized.

Figure 12: Selecting Model

4.2.2 Filling Model

The generation process is the same to 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 13). If the option is correct, the answer will be credited, otherwise the answer will be penalized.

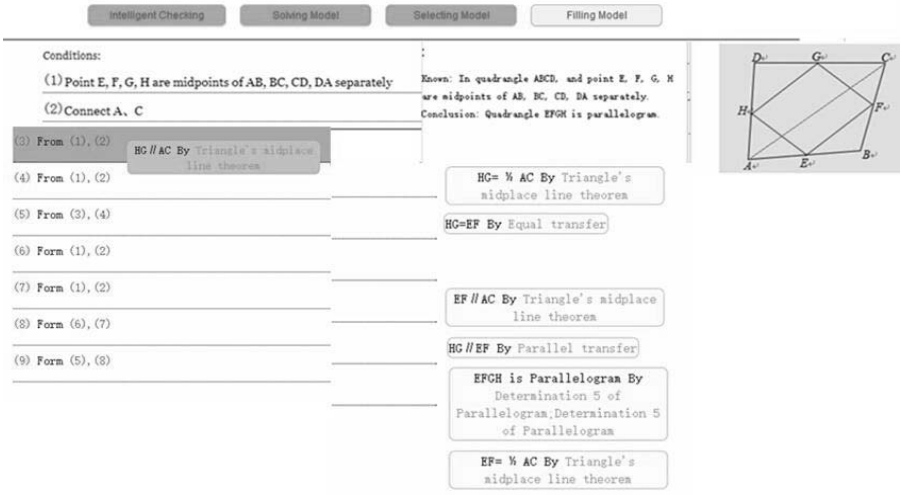


Figure 13: Filling Model

4.2.3 Solving Model

In this model, students can input the problem solving process freely step by step (as is shown in Figure 14), according to their own thinking and problem-solving methods.

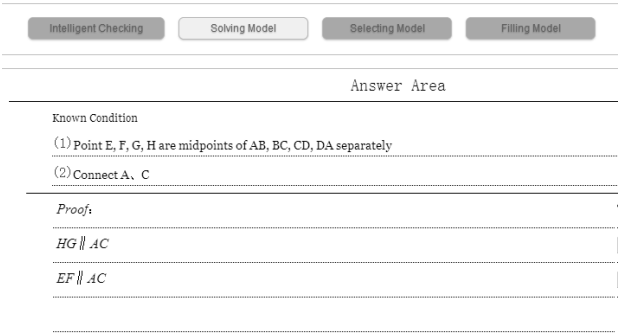


Figure 14: Solving Model

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

5. Conclusion

Testing on subjective question checking for 580 exercises and objective question checking for 100 exercises of junior school mathematics, the accuracy is up to 80%, and the average time is 5 seconds for each one by statistics.

Overall, network structures of problem solving processes generated by selecting model and filling model can allow students have the understanding of the global structures and logical relations. Furthermore, the system can not only carry on objective question checking, but also carry out subjective question checking automatically. What's more, it can not only determine the right or wrong of every step, but also point out the

corresponding error type if possible. Therefore, it is helpful to discover the learning status and the problems of students in time, and then instruct more individualized or recommend more personalized exercises for students adaptively, in order to improve their achievements substantially by the mathematics interactive learning environment.

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) and the Sichuan Provincial Science and Technology Department (Grant No. 2012FZ0120).

References

[1] 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.

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

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.