A framework for automated generation of questions based on first-order logic

Rahul Singhal, Martin Henz, and Shubham Goyal

School of Computing
National University of Singapore (NUS)
Singapore
{rahulsinghal, henz, shubham }@comp.nus.edu.sg

Abstract. In this work, questions are tasks posed to students to help them understand a subject, or to help educators assess their level of competency in it. Automated question generation is important today as content providers in education try to scale their efforts. In particular, MOOCs need a continuous supply of new questions in order to offer educational content to thousands of students, and to provide a fair assessment process. In this paper we establish first-order logic as a suitable formal tool to describe question scenarios, questions and answers. We apply this approach to mechanics (physics) domains in high school education and our experimental results are promising.

1 Introduction and Related Work

Developing assessment material is laborious. Teachers spend countless hours scavenging textbooks and developing original exercises for practice worksheets, homework problems, remedial material and exams. To prevent cheating, many teachers write several versions of each test, multiplying the work required for creating problems. Various standardized tests such as GRE, SAT and GMAT regularly require new questions.

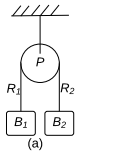
This demand has become more acute with the rising popularity of massive open online courses (MOOCs), in which tens of thousands of students may be enrolled in a course. This massive scale poses a significant technical challenge: creating a large and diverse set of problems of varying difficulty, preventing cheating, and providing new practice problems to students. Hence, there is a need of a software that can quickly generate a large number of questions.

The best developed tutoring systems—JGEX Gao and Lin [2004], Geogebra Geo [2013] and Cinderella Cin [2013] for geometry, ActiveMath Melis and Siekmann [2004] for algebra, and Andes Vanlehn et al. [2005] for physics—do not yet generate questions automatically. A question generation methodology for high school algebra is proposed in Singh et al. [2012], which generates questions similar to a given one, based on a combination of synthesizing and numerical techniques. The generated questions in this approach resemble the given question syntactically and the approach does not consider solution generation.

Alvin et al. [2014] propose an algorithm for generating geometry proof questions for a high school curriculum. The problem generation approach firstly generates a hypergraph that represents all possible proofs over a given pair of user-provided figures and axioms. Later, it systematically enumerates all possible goal sets to find interesting problems. The approach is semi-automated as the user needs to provide a figure to generate questions. Furthermore, Singhal et al. [2014] propose a framework for automated generation of high school geometry questions. However, the scope is limited to the geometry domain.

In this paper, we address the problem of automatically generating new questions and solutions to the satisfaction of a user, based on a specified topic in a given, formally described domain.

Figure 1 (a) pictorial representation of the generated/predefined scenario along with the user input; (b) logical representation of the scenario in the framework; (c) an instance of Question-setting $\mathcal{C}$.



(a)

connected (B_1, R_1),
connected(B_2, R_2),
connected(P, R_1),
connected(P, R_2),
connected(R_1, R_2),
stationary_pulley (P),
acc_block(B_1, A_1),
acc_block(B_2, A_2),
massles_pulley (P),
block (B_1), block (B_2),
rope(R_1), rope(R_2),

(b)

Steps	Before	After
Initialization		$(\mathcal{E}_1, \top, \top)$ where $\mathcal{E}_1$ is the formula in Figure 1b
Extension	$(\mathcal{E}_1, \top, \top)$ where $\mathcal{E}_1$ is the formula in Figure 1b	$(\mathcal{E}_1, \mathcal{I}, \top)$, $\mathcal{I} = equal(T_1, T_2)$ is fact of equal forces on rope added as an implicit fact
Enrichment	$(\mathcal{E}_1, \mathcal{I}, \top)$	$(\mathcal{E}_1 \wedge \mathcal{E}', \mathcal{I}, \top)$, Instantiating masses of blocks and added to scenario $\mathcal{E}' = mass\_block(B_1, M_1), mass\_block(B_2, M_2)$
Commitment	$(\mathcal{E}_1 \wedge \mathcal{E}, \mathcal{I} \wedge \mathcal{H}, \top)$, $\mathcal{H} = acc\_block$ $(B_1, (M_2 - M_1)/(M_2 + M_1) * G)$ added as implicit fact	$(\mathcal{E}_1 \wedge \mathcal{E}, \mathcal{I}, \mathcal{H})$ A question generated from $\mathcal{H}$ is finding the acceleration of block B_1 given the masses of B_1 and B_2 as M_1 and M_2 respectively.

(c)

The proposed framework generates a *scenario* from the user-specified input and then repeatedly adds automatically deduced or generated consistent information, until a question is generated that satisfies the user's requirements.

2 Domain

A *domain* consists of objects, relationships between the objects and the rules that govern their arrangement. We represent a domain by a first-order structure where we describe each object type by a unary predicate and the relevant relationships between objects by non-unary predicates.

The properties of a domain are given by a set of axioms, which are first-order formulas. We illustrate these concepts using mechanics domains.

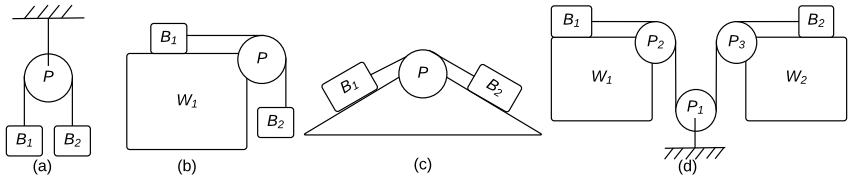
We represent various domain-objects such as pulleys, ropes and blocks by unary predicates such as $pulley, rope$ and $block$, and their relationships by non-unary predicates. For example, the first order formula $connected(P, R)$ expresses the fact that the pulley P and the rope R are connected. For representing quantitative relationships we use real numbers as arguments. For example, $force\_block(B, 10)$ describes the fact that the force acting on the block B is 10 units. Figure 1b shows a partial list of first-order formulas for representing the mechanics domain. We describe the domain rules via axioms.

SWI-Prolog (Version 7.1.2) Schrijvers and Demoen [2004] is used for implementation of predicates. The axioms are represented using Constraint Handling Rules (CHR) Frühwirth and Raiser [March 2011]. In our implementation, we use the CHR library provided by K.U.Leuven, on top of SWI-Prolog. Mechanics domain requires linear equations solver, hence, SWI-Prolog library of CLPR is used.

3 Scenarios

Questions are typically situated in a *scenario*—an arrangement of domain objects that then gives rise to a problem to be solved by the student. We present scenarios by formulas $\exists \bar{X}\ \mathcal{F}(\bar{X})$, where $\mathcal{F}(\bar{X})$ is a formula whose free variables are in $\bar{X}$. We generate a scenario from the domain rules for adding, removing and modifying the objects and their arrangements.

Figure 2 (a) an existing figure scenario; (b) generated by from (a) by adding a fixed wedge; (c) generated from (a) by adding an inclined wedge; (d) generated from (a) by adding multiple wedges, pulleys and blocks.



In mechanics, Figure 2a shows a scenario generated from the domain rules for adding pulleys, blocks, ropes and wedges. Figure 1b represents a partial list of facts and their conjunction forms the scenario as shown in Figure 2a. Figure 2b, c, d and e show the generation of various new scenarios from Figure 2a with the application of domain rules.

4 Question generation framework

Let us assume a given scenario $\exists \bar{X}\ \mathcal{F}(\bar{X})$. A *question* based on this scenario is a more specific formula $\exists \bar{X}\ (\mathcal{F}(\bar{X}) \wedge \mathcal{G}(\bar{X}))$, where $\mathcal{G}(\bar{X})$ is the additional information that the question asks for and that the student needs to provide in an answer. For example, Figure 1c shows a generated fact $\mathcal{H}$ which can be converted to a quantitative question such as finding the acceleration of block B_1.

We propose a framework for producing a state sequence to generate questions where each state is a *question-setting* $\mathcal{C}$, which can be described as: $\mathcal{C} = (\mathcal{E}, \mathcal{I}, \mathcal{H})$, where $\mathcal{E}$ represents explicit facts to be made known to the student, $\mathcal{I}$ represents implicit facts to be hidden from the student and $\mathcal{H}$ represents facts that might constitute a solution to a generated question, of course also to be hidden from the student. In terms of first-order logic, a question-setting $\mathcal{C}$ can be described as a formula, as shown in Equation 1, which is the conjunction of its components $\mathcal{E}, \mathcal{I}$, and $\mathcal{H}$.

$$C = \exists \bar{X}((\mathcal{E}(\bar{X}) \wedge \mathcal{I}(\bar{X}) \wedge \mathcal{H}(\bar{X})) \tag{1}$$

Note that $\mathcal{I}$ and $\mathcal{H}$ may be $\top$. Each component of the state changes along the process of question generation. We discuss the steps used in question generation assisted with an example in mechanics as shown in Figure 1.

Initialization: $(\mathcal{E}_0, \top, \top)$, refers to the scenario generation from the user given topic. $\mathcal{E}_0$ is a scenario representing a set of generated explicit facts. For example, the first row of Figure 1c explains the state transformation in mechanics where we generate a scenario (see Figure 1a) from the user-given topic—pulleys, ropes and blocks. We assume $\mathcal{E}_0$ is sound.

Extension: $(\mathcal{E}, \mathcal{I}, \top) \Rightarrow (\mathcal{E}, \mathcal{I} \wedge \mathcal{I}^{'}, \top)$, refers to the generation of new facts from the predefined domain axioms and explicit facts. For example, the second row of Figure 1c shows an example of generation of a new implicit fact $\mathcal{I}^{'}$. The fact $equal(T_1, T_2)$ implies that the forces acting on the blocks by the rope in the scenario are equal. Our framework generates this fact through application of an axiom that deals with the forces acting on a block by ropes. In first-order logic, this step is represented as $\forall \bar{X}(\mathcal{E}(\bar{X}) \wedge \mathcal{I}(\bar{X}) \rightarrow \mathcal{I}^{'}(\bar{X}))$

Enrichment: $(\mathcal{E}, \mathcal{I}, \top) \Rightarrow (\mathcal{E} \wedge \mathcal{E}^{'}, \mathcal{I}, \top)$, generates new facts $\mathcal{E}^{'}$ and adds them to the existing scenario. The third row of Figure 1c shows an example of an explicit fact $\mathcal{E}^{'}$. The formula $mass\_block(B_1, M_1)$ assigns the mass of block B_1 to M_1. We add these facts to the list of

explicit facts resulting in a specific scenario. In first-order logic, this step is represented as $\exists \bar{X}(\mathcal{E}(\bar{X}) \wedge \mathcal{I}(\bar{X})) \rightarrow \exists \bar{X}(\mathcal{E}(\bar{X}) \wedge \mathcal{I}(\bar{X}) \wedge \mathcal{E}^{'}(\bar{X}))$

Commitment: $(\mathcal{E}, \mathcal{I} \wedge \mathcal{H}, \top) \Rightarrow (\mathcal{E}, \mathcal{I}, \mathcal{H})$, refers to the generation of fact $\mathcal{H}$ which represents the answer of a question. For example, the last row of Figure 1c shows the fact $acc\_block(B_1, (M_1 - M_2) * G/(M_1 + M_2))$ that can be converted to answer of the question of finding the accelerations of blocks B_1 and B_2 given their masses as M_1 and M_2, respectively.

Sound question refers to the question which is internally consistent and has a solution generated from the domain axioms. We propose the following theorem regarding our framework.

Theorem 1. *Any algorithm that starts with an initialization step and generates a sequence of extension, enrichment and commitment steps generates sound questions.*

Each step is either generating implicit facts with the help of domain axioms, adding explicit facts with the help of a function or adding a question fact. The theorem is proven by induction on the number of steps required to generate a question. The initial question setting is sound and each step constitutes a sound transformation in first-order logic.

Along with the question, we generate an answer from the fact $\mathcal{H}$. The answer of a semantic question is generally a proof and hence requires a set of axioms. However, the answer of a quantitative question is the exact value that the question asks for. If c is the answer for a question that asks for the value of X_i, then $\mathcal{H}$ has the shape $X_i = c$ and answer becomes $\exists \bar{X}(\mathcal{S}(\bar{X}) \wedge X_i = c)$

5 Future Work

In this paper, we propose a notion for describing a domain to generate user-desired questions and solutions and described a promising tool for our framework. Future work can be carried in various directions. One major drawback of this approach would be the inability of the framework to handle degeneracy conditions across domains. For example, the framework cannot check the correctness of the function for adding a pulley to a given scenario. Other improvements would involve conducting an experiment in which the teachers would be asked to differentiate between the system generated problems from existing online and textbook questions.

References

Cinderella geometry tool (Jan 2013), `http://www.cinderella.de/tiki-index.php`

Geogebra geometry tool (Jan 2013), `httpS://www.geogebra.org`

Alvin, C., Gulwani, S., Majumdar, R., Mukhopadhyay, S.: Synthesis of geometry proof problems. In: AAAI (2014)

Frühwirth, T., Raiser, F. (eds.): Constraint Handling Rules: Compilation, Execution, and Analysis (March 2011)

Gao, X.S., Lin, Q.: Mmp/geometer asoftware package for automated geometric reasoning. In: Winkler, F. (ed.) Automated Deduction in Geometry, pp. 44–66. Lecture Notes in Computer Science (2004)

Melis, E., Siekmann, J.: Activemath: An intelligent tutoring system for mathematics. In: Artificial Intelligence and Soft Computing - ICAISC 2004. Lecture Notes in Computer Science (2004)

Schrijvers, T., Demoen, B.: The k.u.leuven chr system: Implementation and application (2004)

Singh, R., Gulwani, S., Rajamani, S.: Automatically generating algebra problems. In: AAAI (2012)

Singhal, R., Henz, M., McGee, K.: Automated generation of geometry questions for high school mathematics. In: Sixth International Conference on Computer Supported Education. (2014)

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R.H., Taylor, L., Treacy, D.J., Weinstein, A., Wintersgill, M.C.: The andes physics tutoring system: Five years of evaluations. In: Proceedings of the Artificial Intelligence in Education Conference (2005)