Applications of Computer Algebra – ACA 2019
Montréal, Canada | July 16-20, 2019
École de technologie supérieure

The Modelisation of the Possible Proofs for High School Geometry Problems in the Tutoring Software QED-Tutrix

Ludovic Font*[1]*, Michel Gagnon*[1]*, Nicolas Leduc*[1]*, Pascal-Alexandre Morel*[1]*,
***Philippe R. Richard*[2]*, Michèle Tessier-Baillargeon*[2]** [nicolas.leduc@polymtl.ca]

[1] École Polytechnique de Montréal, Montréal, Canada
[2] Université de Montréal, Montréal, Canada

1 Context

The intelligent tutoring system QED-Tutrix [1,2] aims at providing an environment in which a high school student can solve proof problems in geometry, and at helping the student during the whole resolution process. This is done thanks to a tutoring engine that reads the interaction of the student with the software interface, infers the proof he seems to be working on, and provides him advices to complete his proof if such help is needed.

A crucial part of this engine is a structure in which all the possible proofs (accessible to a high school student) for the problem are stored. This way, it becomes possible to anticipate what the student is likely to attempt to do next, and therefore to help him in a relevant way. In this paper, we present this structure, called the HPDIC graph. Details concerning its functionality are presented in Section 2, and we explain in more depth how the graph is used in QED-Tutrix in Section 3.

2 HPDIC graph representation

To represent a mathematical proof as a computer structure, we must first define precisely what we consider to be a "proof" in the context of high school geometry problem resolution. Our definition, based on the one proposed by Duval [3] is built around the concept of inference. An inference can be seen as an atomic sentence, composed of some premises (ABC is a triangle, and the lengths of line segments [AB] and [AC] are equal), the invocation of a justification in the form of a definition, property or theorem (a triangle with two equal sides is isosceles), and the inferred result (ABC is isosceles). Since the result of an inference can be used as a premise for another, we can build a proof by chaining inferences, starting from the hypotheses of the problem and reaching its conclusion.

The structure of such an inference chain is perfectly suited to be stored in a graph, as has been done in works going as further back as 1984 [4]. Therefore, we define the HPDIC graph (**H**ypotheses, **P**roperties, **D**efinitions, **I**ntermediate results, **C**onclusion) as follows. First, let us consider one proof (i.e., one chain of inferences) for a problem. Each inference is represented by a subgraph: one node for each premise of the inference, one for the justification, and one for the result. These nodes are then linked following two rules : the premises of an inference are linked to the justification representing it, and that justification is linked to its inferred result.

Since the result of an inference can be used as a premise in another one, we merge identical intermediate nodes. These rules create a graph from the hypotheses of the problem to its conclusion. Then, since each proof of a problem uses a subset of the same set of hypotheses, reaches the same conclusion, and uses a subset of the same set of intermediate results, it is possible to put together all the inferences used in **any** proof of a given problem in a graph that we name the HPDIC graph of the problem. This process is illustrated in Figure 1. Figure 1a shows a simple inference. In Figure 1b, this inference (in bold) is used as a step for a complete proof, and, in Figure 1c, this proof (in bold) is merged with another proof, creating the complete HPDIC graph for this fictive problem.

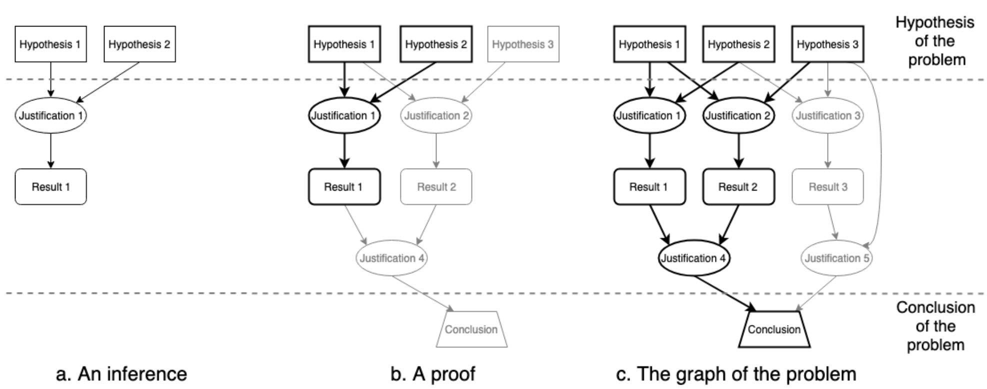

Figure 1: The process of constructing the HPDIC graph

The strength of this representation is twofold. First, it provides a representation of proofs that mirrors the work of high school students. This fundamental requisite is usually not fulfilled by automated theorem proving, since popular methods use intermediate representations, such as the translation of the problem into a system of equations, that is solved by an algorithm [5, 6]. This process provides mathematically valid proofs, but those are also completely out of the scope of high school mathematics education. Furthermore, the broad goal of automated theorem proving is to provide exactly one proof of the theorem, whereas we require the creation of all proofs accessible at a high school level. Second, the structure is very flexible and allows the representation of proofs that use any kind of properties, ranging from small, precise demonstration steps to advanced theorems. This flexibility is crucial when we consider the constant variations in the educational referential. Indeed, the properties that a student is allowed to use change depending on many factors, such as the position in the curriculum, the subject the teacher is emphasizing at the moment, or even the teacher's personal preferences and habits.

3 Uses of the HPDIC graph

During the resolution of a problem by a student in QED-Tutrix, the HPDIC graph is used as a referential to identify the proof that the student seems to be working on. This is done by tagging in the graph each proof element (property or result) entered by the student. Then, an algorithm finds out which proof among all the possible ones is the most advanced, calculated as a percentage of the tagged elements among all the elements used for the proof. This information is recalculated each time the student submits a new element.

Then, using this knowledge, the tutor engine can find out which elements are missing for the student to complete their proof. The tutor is therefore able to guide him/her toward these missing elements. These processes of tagging on the graph and providing messages to help the student are detailed in the work of Nicolas Leduc [1].

In the initial version of QED-Tutrix, the HPDIC graphs were constructed manually by an expert in mathematics education. This process is explained in her work [2]. In particular, the solutions proposed in the HPDIC graph have been validated by several experimentations in class. Five diverse problems were implemented this way, with the aim of encompassing a large number of mathematical concepts, indicating that the HPDIC graph structure is indeed appropriate to represent proofs used in a real classroom context. To improve the scope of QED-Tutrix, we are currently working on a tool to automatically generate the HPDIC graphs for any given problem [7].

Keywords
Proofs, Modelisation, Tutor software

References
[1] N. Leduc, *QED-Tutrix : système tutoriel intelligent pour l'accompagnement d'élèves en situationde résolution de problèmes de démonstration en géométrie plane*. Ph.D. thesis, École polytechnique de Montréal, 2016.

[2] M. Tessier-Baillargeon, *GeoGebraTUTOR : Développement d'un système tutoriel autonome pourl'accompagnement d'élèves en situation de résolution de problèmes de démonstration en géométrie plane etgenèse d'un espace de travail géométrique idoine*. Ph.D. thesis, Université de Montréal, 2015.

[3] R. Duval, *Structure du raisonnement deductif et apprentissage de la demonstration*. Educational Studies in Mathematics, 1991.

[4] D. Gaud; J. P. Guichard, Apprentissage de la démonstration. *Petit x* **vol. 4**, 5–25, 1984.

[5] B. Buchberger, Applications of Gröbner bases in non-linear computational geometry. *Trends in computer algebra*, 52–80, 1988

[6] H. WU, An elementary method in the study of nonnegative curvature. *Acta Mathematica* **vol. 142**, 57–78, 1979.

[7] L. FONT; P.R. RICHARD; M. GAGNON, Improving QED-Tutrix by Automating the Generation of Proofs. In *Proceedings 6th International Workshop on Theorem proving components for Educational software, ThEdu@CADE*, Pedro Quaresma and Walther Neuper, 38–58. Gothenburg, Sweden, 2017.