PROOF GRANULARITY AS AN EMPIRICAL PROBLEM?

Marvin Schiller
German Research Center for Artificial Intelligence (DFKI), Bremen, Germany
Marvin.Schiller@dfki.de

Christoph Benzmüller
International University in Germany, Bruchsal, Germany
Articulate Software, USA
c.benzmueller@googlemail.com

Keywords: proof tutoring, granularity, machine learning

Abstract: Even in introductory textbooks on mathematical proof, intermediate proof steps are generally skipped when this seems appropriate. This gives rise to different granularities of proofs, depending on the intended audience and the context in which the proof is presented.We have developed a mechanism to classify whether proof steps of different sizes are appropriate in a tutoring context. The necessary knowledge is learnt from expert tutors via standard machine learning techniques from annotated examples. We discuss the ongoing evaluation of our approach via empirical studies.

1 INTRODUCTION

Our overall motivation is the development of an intelligent tutoring system for mathematics. Our particular interest is in flexible, adaptive mathematical proof tutoring. In order to make progress in this area it is important to reduce the gap between the existing formal domain-reasoning techniques and common mathematical practice. In particular, the step size (granularity) of reasoning employed in proof assistants and automated theorem provers often mismatches the step size of human-generated proofs. This hampers their usability within a mathematical tutoring environment. For example, when the theorem prover Otter (McCune, 2003) was used in the EPGY learning environment for checking student-generated proof steps, it sometimes verified seemingly large student steps easily, whereas other, seemingly trivial steps were not verified within an appropriate resource limit (McMath et al., 2001). This criticism applies foremost to machine-oriented theorem proving systems, for example, systems based on fine-grained resolution or tableaux calculi. Techniques and calculi that are apparently better suited in this context include, for example, tactical theorem proving (Gordon et al., 1979), hierarchical proof planning (Bundy et al., 1991; Melis, 1999), assertion level theorem proving (..., 2005), (super-)natural deduction (Wack, 2005), and strategic proof search (Sieg, 2007). Such techniques reduce the amount of unnecessary technical details in the generated proofs, support the application of sequences of inference steps as tactics or even the direct application of entire lemmata in single inferences steps (assertion level theorem proving). However, the question remains how large a proof step shall or may be within a tutoring context (comprising a particular proof problem, the didactic goal, and the (assumed) prior knowledge of the student) and how many and which intermediate reasoning steps may be performed implicitly.

To investigate this issue we have analyzed a corpus – the DIALOG corpus (..., 2006) – of tutorial dialogs on proofs. Exploiting assertion level proof search (where each inference is justified by a mathematical fact, such as a definition, theorem or a lemma) the proofs in this corpus have been reconstructed and represented formally in the mathematical assistant system OMEGA (..., 2005). The analyzed students' proof steps generally corresponded to one, two or three assertion level proof steps and very seldomly to four or more. This provides evidence that the step size of assertion level proof comes quite close to the proof step sizes observed in the experiments. However, often combinations of single assertion applications are preferred – even in very elementary proofs.

An example (partial) proof is presented in Figure

1 We assume $(y,x) \in (R \circ S)$ 1 and show $\underbrace{(y,x) \in S^{-1} \circ R^{-1}}_{\Gamma}$.
2 Therefore, $(x,y) \in R \circ S$
3 Therefore, $\exists z$ s.t. $(x,z) \in R \wedge (z,y) \in S$

4 (a) Therefore,
$\exists z$ s.t. $(z,x) \in R^{-1} \wedge (z,y) \in S$
Tutor: "too small"

$$\dfrac{\exists z : (z,x) \in R^{-1} \wedge (z,y) \in S \vdash \Gamma}{\exists z : (x,z) \in R \wedge (z,y) \in S \vdash \Gamma}\ Def.Inv.$$

4 (b) Therefore,
$\exists z$ s.t. $(z,x) \in R^{-1} \wedge (y,z) \in S^{-1}$
Tutor: "appropriate"

$$\dfrac{\dfrac{\exists z : (z,x) \in R^{-1} \wedge (y,z) \in S^{-1} \vdash \Gamma}{\exists z : (z,x) \in R^{-1} \wedge (z,y) \in S \vdash \Gamma}\ Def.Inv.}{\exists z : (x,z) \in R \wedge (z,y) \in S \vdash \Gamma}\ Def.Inv.$$

4 (c) This finishes the current part of the proof.
Tutor: "too big"

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \Gamma(closed)}{\exists z : (z,x) \in R^{-1} \wedge (y,z) \in S^{-1} \vdash \Gamma}\ Def.\circ}{\exists z : (z,x) \in R^{-1} \wedge (z,y) \in S \vdash \Gamma}\ Def.Inv.}{\exists z : (x,z) \in R \wedge (z,y) \in S \vdash \Gamma}\ Def.Inv.$$

Figure 1: Proof sample (for the proof problem $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$, where $^{-1}$ denotes relation inverse and $\circ$ denotes relation composition) with three alternatives for the fourth step, together with tutor's granularity rating (taken from the ongoing experiments), and (partial) assertion level proof reconstruction (as sequent trees).

1: the student has already performed steps (1)-(3) and three alternatives, 4(a)-4(c), for the next step are investigated. They have been annotated (cf. Section 4) by a mathematician concerning their step size. Below these alternatives we outline parts of the corresponding assertion level proofs. Note that the step consisting of only one assertion level inference has been annotated as too small. Here the different classes of step size coincide with different lengths of the associated assertion level reconstructions. Our hypothesis is that such a coincidence generally exists – clearly, not as simple as here – and that we can learn and represent it and exploit it for intelligent proof tutoring. To confirm this hypothesis we currently perform an empirical study which we discuss in this paper.

In Section 2 we present our modeling technique for classifying the steps size of proof steps in a tutoring context. Our approach uses data mining techniques to generate models from samples of proof steps which have been annotated by human experts. Our system can be used in the diagnosis of student steps (to detect whether a student proceeds with unusually small or unexpectedly large steps (..., 2008)), or for the presentation of proofs at a particular level of detail (..., 2009). In order to facilitate and support the collection of annotated sample proofs, we have developed a dedicated, new system environment, which is presented in Section 3. We report on an ongoing empirical study using this new environment in Section 4. Key questions of this study are: (i) How well can we model the judgments of the expert? (ii) How much do judgments differ among various experts? (iii) How well do learned models transfer to other domains? (iv) What are (empirically) relevant properties for classifying the step size of proof steps? We present a summarizing discussion of our approach in Section 5.

2 GRANULARITY AS A CLASSIFICATION PROBLEM

As the basis of our approach to granularity, we hypothesize what properties of compound proof steps[1] may be relevant to judge about their perceived step size. As a result of reviewing our corpus of proofs (..., 2006), we identified the following key features (among others):

- How many different concepts (mathematical facts, such as a particular definition or theorem) are involved within the same compound step? (feature *concepts*)
- How many (assertion level) inferences does a compound step correspond to? (feature *total*)
- are the employed concepts mentioned explicitly? (feature *verb*)
- Is the student familiar with the employed concepts of the compound step? (feature *mastered/unmastered*)
- What theories do the employed concepts belong to (e.g. naive set theory, algebra, topology)? (features *settheory, algebra, topology, etc...*)

Respective feature observations can easily be extracted from assertion level proofs[2]. We also employ a simple student model to keep track of the concepts the student is (presumably) already familiar with.

We treat the decision whether a particular step is of appropriate granularity as a classification problem. Given the properties of a particular step (as a collec-

[1] We use the notion *compound proof step* in the following for any proof step, including those that can be decomposed into the application of several individual inferences - which is not the case for (atomic) proof steps such as single natural deduction inference applications.

[2] Even though the approach is generally not limited to assertion level proofs, we use this proof representation in our study for convenience.

1. total$\in \{0,1,2\} \Rightarrow$ “appropriate”
2. unmastered$\in \{2,3,4\} \wedge$ relations$\in \{2,3,4\} \Rightarrow$ “step-too-big”
3. total$\in \{3,4\} \wedge$ relations$\in \{0,1\} \Rightarrow$ “step-too-big”
4. unmastered$\in \{0,1\} \Rightarrow$ “appropriate”
5. _ $\Rightarrow$ ”appropriate”

Figure 2: Sample rule set generated using C5.0 on sample data. Rules are ordered by confidence for conflict resolution.

tion of its features) we use a classifier (a mapping of a feature vector to class labels) to assign one of the three labels *appropriate*, *step-too-big* and *step-too-small* to it. As classifiers, we use rule sets, which are learned from annotated samples. As an example of such a rule set, consider Figure 2. All our features are numeric, e.g., *unmastered* counts the number of concepts we assume the student is not yet familiar with, *total* counts the number of assertion level inference applications, etc. For example, the proof step 4 (b) in Figure 1, which results in a feature vector (concepts:2, total:1, verb:false,mastered:0,unmastered:1,relations:1, ...) is assigned the label *appropriate* via the rule set in Figure 2 (the first rule fires). Such rule sets can be generated from annotated samples via data mining tools, such as C5.0[3].

3 SYSTEM ENVIRONMENT FOR EMPIRICAL PROOF GRANULARITYSTUDIES

The DIALOG corpus collected data from proof-tutoring dialogs. In these dialogs human tutors (mathematicians) were asked to judge the step size of each student proof step, resulting in a corpus with granularity annotations. This was then used for evaluating the classification approach outlined above, by using data-mining techniques. However, it became apparent that for the in-depth study of granularity, more focused studies are needed since (i) both the students and the wizards were experimental subjects, and the resulting interactions were more geared towards the identification of specific phenomena rather than a controlled experiment, and (ii) both parties were allowed to use natural language freely, which resulted in a large variety of surface realizations of proof steps, often including meta-cognitive comments, which may have had an influence on the judgments of the tutors. The idea of our new environment is to better control the parameters pertaining to the student, in order to more accurately observe their effects on the judgments of the tutor. Therefore, we simulate the student, using: (i) assertion-level proof search in OMEGA, (ii) pattern-based generation of simple natural-language output, (iii) randomization of proof step output (producing compound steps of random size, counting assertion level inferences, and randomizing whether concepts names are explicitly named, or only the resulting formulae are displayed), (iv) automatic collection of all relevant data, including the proof step output, the names of the employed assertion level inferences, the corresponding granularity features, and the corresponding granularity judgments from the tutor.

[3]Data Mining Tools See5 and C5.0: `http://www.rulequest.com/see5-info.html`

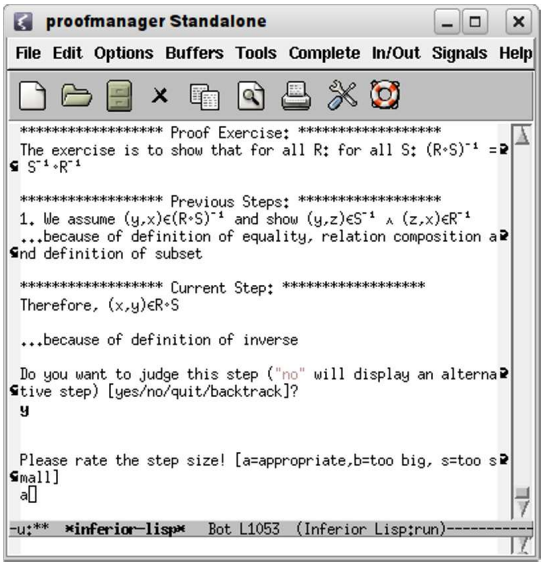

Figure 4: The data collection environment interface

The expert providing the granularity judgments uses the interface in Figure 4. It presents the proof step output and collects the expert judgments. The expert may deny the judgment for a particular step, in which case a different option is presented. When combining several inference steps to a compound step, only inference steps of the same direction (either forward, or backward) are combined, a phenomenon which we clearly observed in the DIALOG corpus. Figure 3 shows a sample of collected data.

4 EMPIRICAL STUDY ON GRANULARITY

Our approach to granularity relies on two assumptions which we investigate empirically:

- We assume that we need an adaptive approach

Proof Step Output	Inferences	Granularity Feature Vector	Judgment
We assume $(y,x) \in (R \circ S)^{-1}$ and show $(y,x) \in S^{-1} \circ R^{-1}$...because of definition of equality and definition of subset	Def.$\subseteq$, Def.=	hypintro:1, total:2, concepts:2, verb:1, ...	appropriate
Therefore, $(x,y) \in R \circ S$	Def. Inv.	hypintro:0, total:1, concepts:1, verb:0, ...	appropriate
Therefore, $\exists z$ s.t. $(x,z) \in R \wedge (z,y) \in S$...because of relation composition	Def. $\circ$	hypintro:0, total:1, concepts:1, verb:1, ...	appropriate

Figure 3: Sample of the data collected in our study.

to granularity, which learns from human experts. The experiments reported by (..., 2006) hinted at the possibility that experts do not always agree with respect to what step size they consider appropriate. We want to compare samples from different experts with tutoring experience and examine the inter-rater reliability.

- We assume a set of features which we consider relevant for classifying granularity (currently around twenty features plus indicator features for each theory and each concept). Our goal is to evaluate (i) which features are most salient, and (ii) what features are potentially relevant?

Therefore, we perform an empirical study, where several mathematicians with tutoring experience judge proof steps presented to them via our data collection environment. Exercises are taken from the fields of naive set theory, relations (such as our running example), and topology. The first experiment session, where a mathematician judged 135 proof steps using our environment, took place recently. However, we plan with two or three more experts, so that differences in their judgment can be examined. The mathematical experts are not instructed about assertion level proofs and the features we use in our classification before completion of the experiment, to avoid an artificial bias. Afterwards, we discuss our approach with the experts to obtain additional feedback.

5 DISCUSSION

We have sketched a flexible, adaptive approach for modeling and assessing proof step granularity. It is based on the collection of empirical data from experts, which is then modeled via artificial intelligence and data mining techniques. In our experiment we have avoided asking the experts explicit questions about how they would go about judging step size by building a hypothetical model using machine learning. It is debatable, whether it is more appropriate to copy current mathematical practice from practitioners (e.g., using the learned classifiers in an intelligent tutoring system) or to establish an explicit best practice by directly researching the different cognitive dimensions involved in judging proof step granularity. We consider our work and our system environment as a fruitful first step in both directions and welcome any feedback at the conference.

REFERENCES

Bundy, A., van Harmelen, F., Hesketh, J., and Smaill, A. (1991). Experiments with proof plans for induction. *J. Autom. Reasoning*, 7(3):303–324.

Gordon, M. J. C., Milner, R., and Wadsworth, C. P. (1979). *Edinburgh LCF*, volume 78 of *LNCS*. Springer.

McCune, W. (2003). Otter 3.3 reference manual. www.mcs.anl.gov/AR/otter/otter33.pdf.

McMath, D., Rozenfeld, M., and Sommer, R. (2001). A computer environment for writing ordinary mathematical proofs. In *Proc. of LPAR 2001*, volume 2250 of *LNCS*, pages 507–516. Springer.

Melis, E. (1999). Knowledge-based proof planning. *Artificial Intelligence*, 115:494–498.

Sieg, W. (2007). The apros project: Strategic thinking & computational logic. *Logic Journal of the IGPL*, 15(4):359–368.

Wack, B. (2005). *Typage et deduction dans le calcul de reecriture*. PhD thesis, Universite Henri Poincare Nancy 1.

... (2005). < hidden reference>.

... (2006). < hidden reference>.

... (2008). < hidden reference>.

... (2009). < hidden reference>.