

# PG Tips: A Recommender System for an Interactive Theorem Prover



# PG Tips: A Recommender System for an Interactive Theorem Prover

Alison Elizabeth Mercer * Alan Bundy Hazel Duncan
David Aspinall

School of Informatics, University of Edinburgh
http://dream.inf.ed.ac.uk/projects/RecommenderSystem

**Abstract**

Interactive theorem provers require input from users to guide the proof process. Some theorems can be complicated, making it difficult to decide which direction to take at a specific point within a proof. PG Tips is a recommender system that has been incorporated into the theorem prover Isabelle's graphical user interface, Proof General, in order to assist users.

Recommender systems are used, in a variety of situations, to provide predictions based on information supplied by a user. In this case, PG Tips is used to suggest possible proof steps based on the analysis of previous proofs. It is hoped that the creation of such a system will help users in finding proofs and accelerate the proof authoring process.

## 1 Background

There are various proof assistants available, such as Isabelle, Coq, HOL and Otter, which allow users to prove mathematical theorems. Isabelle [6] is a commonly used interactive theorem prover that works with a variety of logics. Like other interactive theorem provers, Isabelle requires users to enter commands in order to direct the proof. Such commands are constructed using inference rules that are defined within existing theories, provided by Isabelle. If a command is applicable it is applied to the current proof state in order to create a set of subgoals. The type of rule used within a command indicates what parts of the formula are altered. Further commands are then used to solve the subgoals and therefore prove the theorem. There are various types of rules available that manipulate the formula in different ways.

*This paper is based on the system developed within Alison Mercer's undergraduate honours project report. The system was created using the data provided by Hazel Duncan's PhD work.

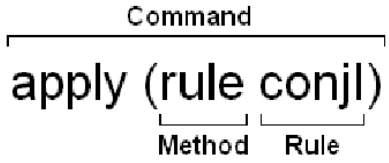


Figure 1: Example of a Command

Within a command (Figure 1), each rule has an associated method which controls how it is applied. In this paper, the method and rule are collectively referred to as the *proof step*. Isabelle attempts to ease the proof process by providing commands that allow parts of the proof to be automated. Although the inclusion of such functionality assists users, it does not remove the issues involved with deciding which commands to apply at each stage of the proof. Incorporating a recommendation option into Isabelle, which suggests possible steps the user could take in order to progress within a proof, could further aid user interaction.

Isabelle can be accessed via a shell window or user interface. Proof General [2] is the most commonly used interface. It supports the use of Isabelle as well as a number of other interactive theorem provers, including Coq, PhoX and LEGO. Isabelle users often use Proof General as it provides them with the ability to easily write and maintain their proofs. It is therefore useful to consider the adaptation of Proof General when looking at incorporating a new feature into Isabelle.

For many years, recommender systems [1] have been used to make suggestions based on information gained from previous users. The first recommender system, Tapestry [4], was a mail system with the ability to manage received electronic documents using opinions generated by those who had already read them. It worked by searching the feedback related to a specific document in order to ascertain whether or not it was relevant to the owner's interests. The design of this system follows a technique known as collaborative filtering, which requires gathering information from other users so as to provide a recommendation. Suggesting possible steps to Isabelle users can be accomplished based on such a method. Details regarding patterns that are located in previous proofs can be gathered and searched for instances that are related to the user's current proof.

Data-mining techniques such as Variable Length Markov Models [3] are used to generate such patterns. They are able to identify common sequences that occur within various types of data. Each pattern found is given a probability, which states the likelihood of that sequence appearing. Markov Models are useful as they can be used to predict the probability of a subsequence occurring based on prior information. In situations where the length of patterns vary it is necessary to use Variable Length Markov Models.

Hazel Duncan, a PhD student at the University of Edinburgh, has already used data-mining

in this way. Her system, IsaNewT [5], automatically forms tactics from a large number of proofs in Isabelle. This initially involves using Variable Length Markov Models to identify common sequences that occur within a set of chosen proof corpora. It works by searching previous proofs and calculating the number of times that each sequence appears within them. The likelihood of each sequence occurring is recorded and updated each time it is found. Each sequence therefore contains a set of proof steps where each step has an associated probability that states the likelihood of a step occurring based on the appearance of the previous steps in the sequence. The common sequences identified during this phase are suitable to use within a recommender system for Isabelle as they provide the patterns required in order to make a recommendation. To the best of our knowledge, there have been no previous attempts to implement a recommender system within existing theorem provers.

# 2 Overview of PG Tips

The aim of incorporating a recommendation option into the interactive theorem prover Isabelle is to further assist users in proving mathematical theorems. A recommendation that suggests possible proof steps, by comparing the steps taken by a user with those contained within previous proofs, should be provided on request. To gain a suitable recommendation PG Tips searches the relevant common sequences generated by IsaNewT for instances containing all of the steps the user has taken. The system can then be used via Proof General.

## 2.1 Proof General Interface

In order to incorporate the recommender system into Proof General, the interface was altered to include a recommendation button (right-most button in Figure 2).

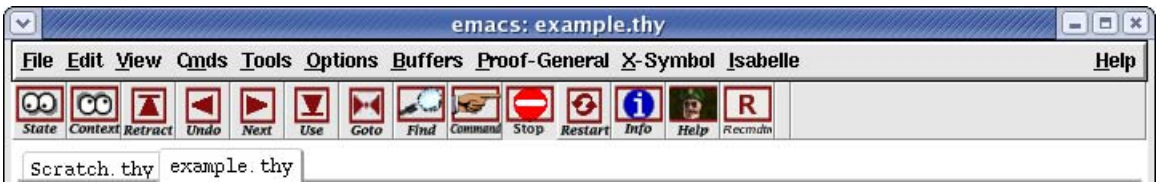


Figure 2: Toolbar of Proof General Interface

When pressed, the commands taken so far by the user are gained from within Proof General and sent to PG Tips so it can provide a recommendation.

## 2.2 Recommender System

Once PG Tips has abstracted the required information from that sent by Proof General it must search all of the relevant common sequences for instances where the steps gained are the initial steps. This allows the system to find sequences that contain the steps taken by the user and provide a recommendation suggesting steps that previous users took next. When

run, IsaNewT produces a number of files that each include patterns derived from specific theories which are defined within Isabelle. PG Tips has been implemented to use the sequences generated from the Isabelle/HOL theory *Main* which defines a number of sub-theories including predicate logic, set theory and inductive data-types. However, it has also been set up so that if it was to receive the theory name the user's theory is based on, it would be able to search the associated file generated by IsaNewT and return recommendations which would be better targeted to that particular theory.

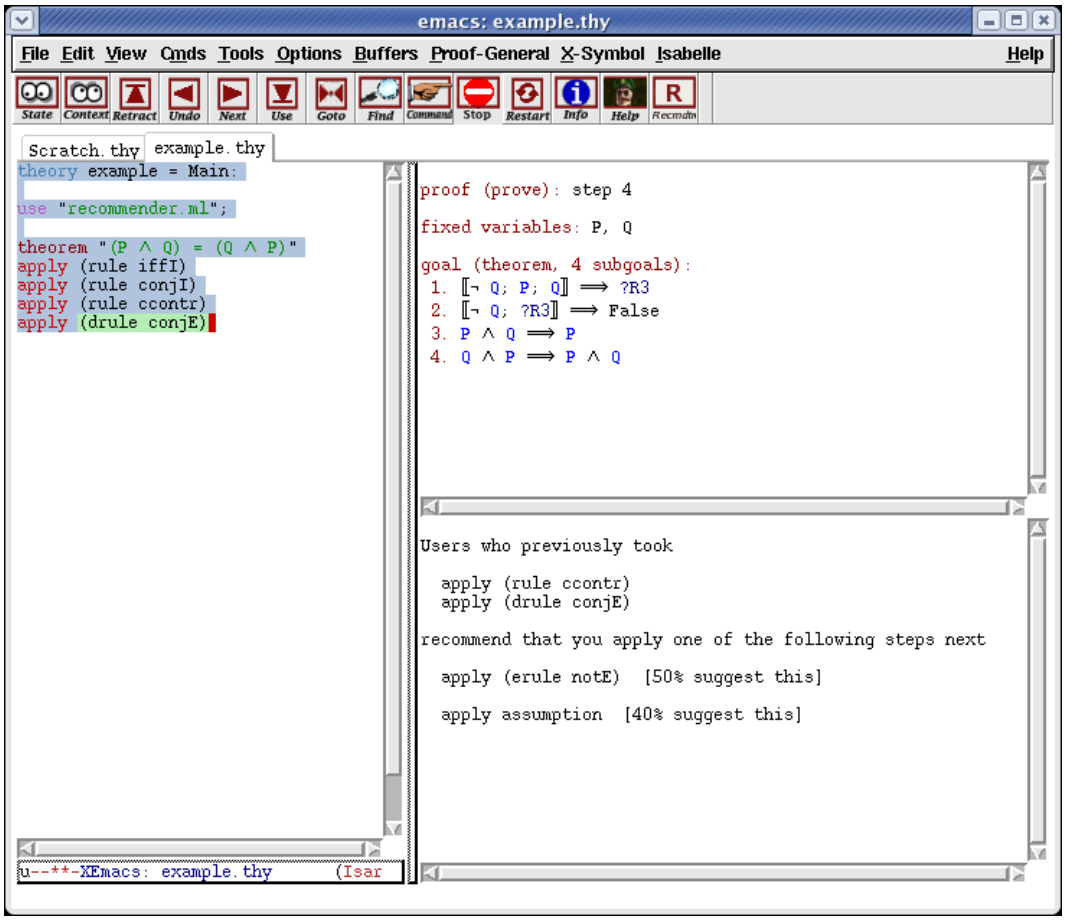


Figure 3: Example of Interaction with the Recommender System

When a matching sequence is found, the proof steps that follow those taken by the user are displayed along with the associated probability of those steps occurring, as shown in Figure 3. In the current system, to accept suggested steps users must copy and paste the provided recommendation into their proof script. The Next button, situated within the toolbar, can then be used to apply the recommended step to the proof state. Future work will look at easing this process by automatically inserting and applying a recommended step when it is selected by the user. It may be the case, that more than one match is

found during the search. In such a situation, the system returns the steps with the highest likelihood of occurring. This is logical as the user will want the recommendation that is most likely to assist them in their proof. PG Tips has been designed to return the top three steps with the highest probability of occurring. Such functionality offers users more options but remains below the limit where a user may become confused by the number given.

In the situation where a search completes with no results, it is necessary to take further action. If the steps provided by the user do not gain a recommendation then the first step is dropped from those given and the search repeated on the new set of steps. For example, when a user takes steps A, B then C and there are no common sequences that contain these steps the search is repeated with steps B and C. This process continues until a match is found or there are no steps left to search with.

# 3 Evaluation Results

A user-based evaluation method was used in order to gain the views of potential PG Tips users. This involved the creation of a questionnaire, which was distributed to ten willing participants with varying knowledge of Isabelle/Proof General. Initially, participants were required to prove a total of six mathematical theorems. They were asked to solve the first three theorems as they would normally, without using PG Tips, and then to use it to help them prove the rest. Asking them to prove both sets of theorems in different ways allowed the participants to gain an idea of how the recommender system affects the proof process. Once the participants completed the interactive evaluation they were asked to answer the series of questions provided within the questionnaire. The questionnaire was created in order to gain specific information regarding certain aspects of the recommender system. The evaluation focused on the quality of advice provided by PG Tips and the benefits of such advice to the user. The aim was to discover if the system supplies recommendations that assist users in their proofs.

It is important that the majority of advice offered by PG Tips is beneficial to the user. In most cases, this would mean that the system suggests a proof step that the user could then employ within their proof. In order to evaluate the quality of the recommendations it is appropriate to look at how often users took the advice that was given to them. It is also relevant to examine whether the system suggested commands that had previously been unthought-of by the user. This must be analysed in respect to whether the user took the advice and if it was beneficial, as it does not necessarily mean that because it was unthought-of that it was useful.
The bar chart shown in Figure 4 illustrates the results gained in respect of the questions about the quality of the recommendations provided by PG Tips. Overall, it can be seen that most participants agree with the assertion that the suggestions generated by the system were beneficial to them. However, the expert user who took part in the evaluation strongly disagreed with this statement. His written comments stated that this was due to the inability of

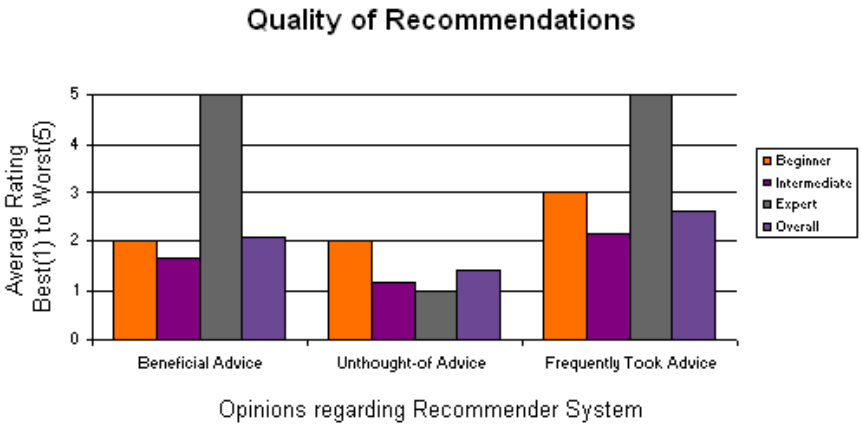


Figure 4: Bar Chart illustrating the Quality of Recommendations

the system to take into account the proof's current goal when providing a recommendation. Although this is a desirable feature it is something that could not be completed within the specified timescale for this project. Based on the results provided by the other participants of the evaluation it is fair to say that in most cases the recommender system does provide beneficial advice without the inclusion of such functionality. While eight users either agreed or strongly agreed that the recommendations were useful, in general, seven agreed with the idea that they frequently took the advice given to them. If further evaluation of this system was to take place it would be useful to set up a means of logging how often users took the advice they were given. On the whole, participants agreed that the recommendations provided by PG Tips made suggestions that they had previously not thought of. It can be observed that in most cases this advice helped the user.

The results gained from the evaluation illustrated that the majority of participants found PG Tips useful. Most of them agreed that the recommendations helped ease the proof process, with one intermediate user stating that the system made Isabelle easier to use as it provided ideas when they were stuck. Many stated that the system was easy to use, with nine participants regarding PG Tips as a useful feature within Proof General that they would use again.

# 4 Further Work

Further work regarding PG Tips is due to commence in the near future.[1] The aim is to incorporate some of the additional features mentioned below in order to enhance the usefulness of the system.

[1]Features mentioned in sections 4.1 and 4.2 are the subject of a current project.

## 4.1 Applicability of Recommendation

Currently, PG Tips provides recommendations based on the common sequences generated by IsaNewT. In some cases, the command that it suggests is not applicable to the current proof state. Adapting the system to check applicability before a recommendation is made would enhance the quality of assistance provided. The inclusion of such a feature would require altering Isabelle to check the applicability of the proof steps gained before displaying them to the user. In cases where the suggested steps cannot be applied Isabelle can then query PG Tips for further recommendations, if there are more available.

## 4.2 Automatic Insertion of Recommendation

The recommendations are displayed within the response window of the Proof General interface. At the moment, if users wish to apply suggested commands they are required to copy and paste the relevant information into their proof and use the Next button to alter the proof state. To ease this process it would be beneficial to incorporate a feature into Proof General that automatically inserts a selected step into the correct position. Proof by pointing has already been implemented in Proof General for the interactive theorem prover LEGO. It provides users with the ability to select specific terms within the subgoals, which they believe are significant within the proof. The proof assistant then automatically executes commands to make progress in the proof, based on the selected term [7]. The concepts employed to implement proof by pointing within Proof General can be used in order to provide the automatic insertion of chosen commands.

## 4.3 Modification of Stored Common Sequences

The advice given by PG Tips is dependant on the quality of sequences generated by IsaNewT. It would therefore be beneficial to dynamically update the likelihood associated with a pattern that is successfully used within a proof. There are cases where the steps recommended by the sequence are applicable but not appropriate within a particular proof. In order to ensure that only successful patterns are updated it is important that they are not modified until the proof has been completed.

## 4.4 Adaptation for other Interactive Theorem Provers

Although PG Tips has been implemented to work with proofs in Isabelle it would be possible to alter it so that it runs with other proof assistants. Since the system has already been incorporated into Proof General it would be appropriate to modify it to support the various proof assistants that can be used via the interface. This includes Coq, PhoX and LEGO. Altering the system to work for each of these should be reasonably straightforward once sufficient research has been conducted in regards to each interactive theorem prover. It is important to note that IsaNewT would also have to be altered in order to gain patterns from previous proofs relating to the prover of choice.

# 5 Acknowledgements

We would like to thank Lucas Dixon for his assistance throughout this project.

For further information regarding PG Tips, please visit the website at
http://dream.inf.ed.ac.uk/projects/RecommenderSystem.

# References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *volume 17 of IEEE Transactions on Knowledge and Data Engineering*, 2005. http://dx.doi.org/10.1109/TKDE.2005.99.

[2] David Aspinall. Proof general: A generic tool for proof development. *volume 1785 of Lecture Notes in Computer Science*, 2000, Springer-verlag. http://proofgeneral.inf.ed.ac.uk/papers/pgoutline.ps.gz.

[3] Y. Singer D. Ron and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning, 25*, 1996. http://www.cs.huji.ac.il/~singer/papers/psa.ps.gz.

[4] Brian M. Oki David Goldberg, David Nichols and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992. http://www.ischool.utexas.edu/~i385d/readings/Goldberg_UsingCollaborative_92.pdf.

[5] Hazel Duncan. *The use of Data-Mining for the Automatic Formation of Tactics*. PhD thesis, University of Edinburgh, 2006. Forthcoming. http://dream.inf.ed.ac.uk/DataMining/.

[6] Lawrence C. Paulson. Isabelle: The foundation of a generic theorem prover. *Automated Reasoning, 5*, 1989. http://www.cl.cam.ac.uk/Research/Reports/TR130-lcp-generic-theorem-prover.pdf.

[7] Thomas Kleymann-Schreiber Yves Bertot and Dilip Sequeira. Implementing proof by pointing without a structure editor. *LFCS Technical Report ECS-LFCS-97-368*, 1997. http://www.lfcs.inf.ed.ac.uk/reports/97/ECS-LFCS-97-368/ECS-LFCS-97-368.ps.