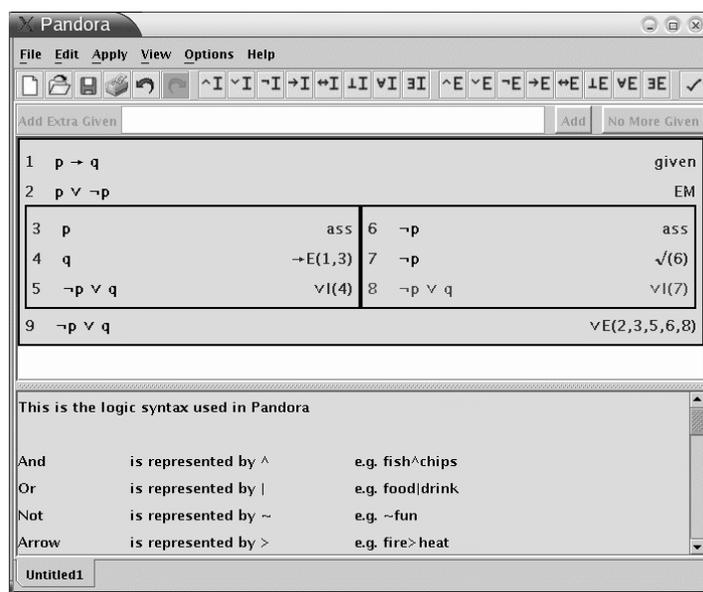


Pandora: Making Natural Deduction Easy

Krysia Broda, Jiefei Ma, Gabrielle Sinnadurai, and Alexander Summers

Department of Computing, Imperial College London
{kb,jm103,apgs,ajs300m}@doc.ic.ac.uk

Abstract. Pandora is a tool for supporting student learning of First Order Natural Deduction. It provides students with instant feedback as every attempt to apply a natural deduction rule is met with either success or a helpful error message. It has a help facility and an interactive context sensitive tutorial. Here we introduce the tool, describe the environment within which it is used at Imperial College and give some evaluation and lessons learned.



1 History and Context

Natural deduction has been taught to first year undergraduates of Computer Science and Joint Mathematics and Computing at Imperial College since 1991. Most students find the high degree of rigour required in formal natural deduction proofs daunting.

Pandora (Proof Assistant for Natural Deduction using Organised Rectangular Areas) is a learning support tool designed to guide the construction of natural deduction proofs. It was first conceived in 1996 and is based on the ‘‘Proof box notation’’ for Natural Deduction as described in [1], which has been the recommended text for the course since it was developed from the course lecture notes. The first version of pandora was written by a final year student using Tcl/Tk in 1996 and was available for students to use in tutorials but was not promoted in lec-

tures. A second version was written in an early version of Java in 1999 by another project student. This second version was more robust and was used for demonstrations in class as well as tutorials. It was available for students to download onto their own machines and promoted vigorously and was consequently widely used. The third and current version is also in Java and was developed as a group project in 2002 and further enhanced by several summer students. It provides the basic functionality together with a help module, a context sensitive tutorial and the facility to export proofs as latex. The tutorial, added in 2004, is the most recent addition. An applet version is available for use via the pandora web site [2] and we hope to make it available for download in the near future. Pandora is now so robust we consider it safe for a small cohort of students to use the tool for an on-line ‘‘driving test’’.

2 Overview of Current Version

Pandora provides learning support to guide students in their construction of a natural deduction proof of a conclusion or goal from given premisses. It allows the user to reason “forwards”, that is, from one or more given formulas deduce another formula using one of the rules, and to reason “backwards”, that is, to reduce one of the current goals to one or more subgoals from which the current goal can be deduced using one of the rules. The rules are all the usual introduction and elimination rules of first order natural deduction with equality plus a few derived rules found useful by more advanced users.

We will explain the usual use of the tool by working through a small example to derive $p \rightarrow q$ from the premiss $\neg p \vee q$. On starting a new proof pandora first requires the premisses and goal to be given. These are typed by the user into a text box. In our example there is only one premiss so we type $\sim p|q$ then click on the “no more given” button. Pandora then requires us to give the goal. We type $p>q$ into the text box. Pandora checks that these are well formed formulas and then displays the initial state of the partial proof with the premisses at the top and goal at the bottom.

```

1   $\neg p \vee q$            given
    $< empty >$ 
2   $p \rightarrow q$        conclusion

```

Notice that we input the symbols for “not”, “or” and “implies” as the plain text characters \sim , $|$ and $>$ but pandora displays them as \neg , \vee and \rightarrow . We experimented with using buttons to represent the symbols but found that plain text was by far the most convenient method for user input and having to learn a second set of symbols for the connectives did not present any problem to our students.

Now for the proof. First we apply the \rightarrow introduction rule backwards by selecting the goal, $p \rightarrow q$, and clicking on the $\rightarrow\mathcal{I}$ button. Pandora displays the new proof state. The box shows the scope of the assumption, p , which is discharged by the $\rightarrow\mathcal{I}$ rule and the conclusion, $p \rightarrow q$, now has the justification $\rightarrow\mathcal{I}(2,3)$.

```

1   $\neg p \vee q$            given
    $< empty >$ 
2   $p$                    ass
    $< empty >$ 
3   $q$ 
4   $p \rightarrow q$         $\rightarrow\mathcal{I}(2,3)$ 

```

Next we would like to apply the \vee elimination rule to split the premiss $\neg p \vee q$ into its two cases. The current goal is q and we want to maintain this so we must use the rule backwards by selecting the goal, q , and clicking on the $\vee\mathcal{E}$ button then clicking on the premiss $\neg p \vee q$. Pandora displays the new proof state.

```

1   $\neg p \vee q$            given
    $< empty >$ 
2   $p$                    ass
    $< empty >$ 
3   $\neg p$                ass
    $< empty >$ 
4   $q$ 
5   $q$                    ass
    $< empty >$ 
6   $q$ 
7   $q$                     $\vee\mathcal{E}(1,3,4,5,6)$ 
8   $p \rightarrow q$         $\rightarrow\mathcal{I}(2,7)$ 

```

In the right hand box we have q as an assumption and q as the goal so we can just use the “tick” rule to note that the goal is proven. So we select the goal q , then click on the \checkmark button, then on the assumption q . Pandora displays the new proof state with the $< empty >$ line removed from the right hand box which is greyed out as that part of the proof is completed.

Pandora displays the new proof state. To save space we will not show it yet, but in the left hand box will make our first application of a rule forwards. We have both p and $\neg p$ in scope so we can deduce \perp by the \neg elimination rule. To do this in pandora we select the $< empty >$ line in the left box, then click the $\neg\mathcal{E}$ button, then click on the two formulas which give the contradiction. Pandora displays the new proof state.

```

1   $\neg p \vee q$            given
    $< empty >$ 
2   $p$                    ass
    $< empty >$ 
3   $\neg p$                ass
4   $\perp$                   $\neg\mathcal{E}(2,3)$ 
    $< empty >$ 
5   $q$ 
6   $q$                    ass
7   $q$                     $\checkmark(6)$ 
8   $q$                     $\vee\mathcal{E}(1,3,5,6,7)$ 
9   $p \rightarrow q$         $\rightarrow\mathcal{I}(2,8)$ 

```

Next we apply \perp elimination backwards from the goal q in the left hand box by selecting the goal then clicking on the $\perp\mathcal{E}$ button. This gives \perp as the new goal. As usual, pandora would display the new proof state.

Finally we use the tick rule again to note that we have \perp and we need to show \perp . The proof is now complete. Pandora removes all the empty

lines and greys out the proof and is intelligent enough to remove the last application of the tick rule which is not needed in the final proof. Note that the line numbers and references to them in the justifications were consistently updated as the proof emerged. Note also that in the completed proof every line has a justification.

1	$\neg p \vee q$		given
2	p		ass
3	$\neg p$	ass	6 q ass
4	\perp	$\neg\mathcal{E}(2,3)$	
5	q	$\perp\mathcal{E}(4)$	7 q $\checkmark(6)$
8	q	$\vee\mathcal{E}(1,3,5,6,7)$	
9	$p \rightarrow q$	$\rightarrow\mathcal{I}(2,8)$	

If at any stage the student tries to apply a rule wrongly they are given an error message.

As well as the usual use of the tool to construct a proof, the help window, the interactive tutorial and the export to latex facility are always available. All the proof states shown above were generated using pandora's export to latex option. The student can also save and reload partially constructed or complete proofs and work on several proofs at once, each in its own proof window. There is an undo button so that students can undo some proof steps and try another direction.

Inside the help window the information is divided up into help pages. There is a page for each rule, a page for each menu and a number of other pages such as "The Logic Syntax used in Pandora" and "Hints for Doing Proofs". It provides three navigation tools: a help tree, which lists all the help pages in a tree structure, an index, which lists all the pages in alphabetical order, and a search tool, which allows you to search for a page by its content.

The interactive tutorial consists of a number of propositional exercises which have been divided into three levels of difficulty: "easy", "medium" and "hard". While the student tries out an exercise, an electronic tutor or "etutor" will aid them. If they get stuck in doing a proof, they can ask for hints from the etutor. Although pandora does not allow a rule to be applied incorrectly it does allow bad decisions to be made. For example, suppose you are given $p \vee q$ and the goal is $q \vee p$. You can apply the \vee introduction rule backwards on the goal to reduce it to q but this would be a bad move because it is not possible to prove q from $p \vee q$. If a student makes such a bad move in the tutorial, the friendly etutor will give the warning "Be careful, one or more

of your goals are not provable!" and if the student selects the "why?" option the etutor will spell out a situation in which the premisses are all true but the goal is false.

3 Use in teaching

When teaching natural deduction to our first year students we start by explaining the propositional natural deduction rules and presenting hand written examples of proofs. We explain how proofs are constructed by applying rules forward from the given data or backwards from the goal. Only after they have seen several proofs and tried a few on paper themselves do we introduce them to pandora. We demonstrate pandora using several of the same examples so the students can focus on how to drive pandora rather than on how to prove the theorems. Over the next few weeks we give the students many exercises, some assessed, and introduce them to the first order and equality rules.

We finish the course with a "driving test" consisting of ten problems which the students have one hour to attempt under exam conditions. The help and tutorial are still available. The proofs are gathered electronically, then checked for correctness and converted to latex by "text to text" command line programs included in the pandora package. We can thus produce a report for each student and a summary of results for their tutors with minimal human intervention. Last year the test problems were:

1. $\vdash (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$
2. $\neg p \vee q \vdash (p \rightarrow q)$
3. $\neg(p \rightarrow q) \vdash p \wedge \neg q$
4. $p \rightarrow (q \rightarrow (r \vee s)), \neg(r \vee t), (s \wedge q) \rightarrow t \vdash q \rightarrow \neg p$
5. $\forall x \neg p(x) \vdash \neg \exists x p(x)$
6. $\neg \exists x p(x) \vdash \forall x \neg p(x)$
7. $\neg \forall x p(x) \vdash \exists x \neg p(x)$
8. $\neg \exists x \exists y (\neg x = y) \vdash \forall x \forall y (p(x, y) \rightarrow p(y, x))$
9. $\forall x (x = a \vee x = b), \neg(a = b), g(a) = b, \forall x \forall y (g(x) = g(y) \rightarrow x = y) \vdash g(b) = a$
10. $\forall x \forall y \forall z (r(x, y) \wedge r(y, z) \rightarrow r(x, z)), \forall x (r(x, a) \vee r(x, b)), r(a, b) \vdash \exists y \forall x r(x, y)$

All exercises done throughout the term including the driving test are essentially formative. The main summative assessment is an end of year written examination.

Natural deduction forms a substantial part of their logic course and prepares them for a course in reasoning about programs given in the second term of the first year and a lab exercise on

automated reasoning using the generic theorem prover Isabelle [3] done in the second year. We do not yet have a tool to support the learning of reasoning about programs but we are working on one. As part of the lab exercise on automated reasoning we encourage the students to first prove theorems using pandora and then to use their proof to guide the construction of a proof in Isabelle. Most students find this approach initially useful but gradually learn to use Isabelle independently.

4 Evaluation

We have tried to evaluate pandora and its use in teaching in several ways. Firstly, we have simply asked the students what they think of it, both verbally and using anonymous feedback forms. Secondly, we have tried comparing performance on the written exam by a cohort who were forced to use pandora and another cohort who were not encouraged to use it. Thirdly, we have electronically recorded detailed logs of the students' use of pandora; essentially every "click" they make is recorded so that we can see in detail how they actually use it.

The verbal and written feedback is generally encouraging and students say and write that they enjoy using pandora and find it useful.

The comparison of performance on the exams did not give the clearcut advantage to pandora users that we hoped for and in fact there was little difference in terms of marks between the two cohorts. There did, however, seem to be a difference in style, namely that those who used pandora were much more at home with using rules backwards and did not make "arbitrary" assumptions which they had no hope of discharging whereas the cohort who did not use pandora mainly reasoned forwards and frequently made arbitrary assumptions. It is hard for us to estimate the extent to which this latter cohort compensated by working harder on hand written proofs. We had feared that pandora users may find it hard to adapt to writing proofs by hand but it turned out that the users were more precise syntactically in their hand written proofs than were the non-users.

Some of the results from analysing the electronic logs came as a surprise and disappointment. We worked hard to develop good help and tutorial facilities but the logs showed that they were rarely used. Indeed, more than half of the

students did not look at either the help or the tutorial at all and none had worked systematically through the whole tutorial. The logs also showed a surprisingly high failure rate in students' attempts to apply the rules. A small number of students had virtually no failures but many had almost as many failures as successes. Analysis of the logs showed that many students were not selecting the $\langle \text{empty} \rangle$ line or a goal line before applying a rule. It seems that the natural instinct is to hit the rule button first and then the lines to which you want to apply it. Comparing the logs for the driving test with those for the previous work we were pleased to observe that, with experience, the proportion of failed rule applications decreased.

The logs yielded detailed information about the types of errors students commonly made for each rule. For example, they frequently tried to apply the $\neg\mathcal{I}$ rule backwards to a formula which was not a negation whereas it was comparatively rare for them to try applying $\rightarrow\mathcal{I}$ backwards to a formula which was not an implication. We believe they were confusing the $\neg\mathcal{I}$ rule with the derived "proof by contradiction" rule.

5 Conclusion

Overall the evaluation has taught us that pandora is well liked and considered useful by the students, but that to improve the learning outcomes we should modify our teaching in a number of ways including:

- give an advertisement for the tutorial and help as part of the initial demonstration
- when demonstrating the individual rules emphasise that either the $\langle \text{empty} \rangle$ or goal line needs to be selected **before** clicking the rule application button
- explain how to avoid what the logs show to be the common pitfalls in applying the rules.

We look forward to making good use of what we have learned from our evaluation to improve pandora's usefulness in future teaching.

References

- [1] K. Broda, S. Eisenbach, H. Khoshnevisan, S. Vickers. *Reasoned Programming*. Prentice-Hall, 1994.
- [2] <http://www.doc.ic.ac.uk/pandora/>
- [3] <http://www.cl.cam.ac.uk/Research/HVG/Isabelle/>