Summary of findings

The relationships between student background, Jape usage and test scores

A quantitative analysis of tests, surveys and logfiles suggests that students' backgrounds appear to have little effect either on how much the program was used, or on how much progress was made with the conjectures in the program. However, on average, the more a student used Jape the higher his or her score in course outcome measures. Moreover, progress in Jape is a significant factor in course performance, even when significant background variables such as gender, degree course, and prior programming experience are taken into account.

A comparison of students' proving behaviour on paper and in ItL Jape

Forward-fixated reasoning

On paper, there was clear evidence of a distinction between behaviour that was fixated on reasoning forwards and behaviour that was flexible about reasoning forwards or backwards. So, for example, a forward-fixated reasoner would want to *make* assumptions when a flexible reasoner would *calculate* assumptions by reasoning backwards. In Jape, however, many forward-fixated reasoners quickly discovered that the usefulness of applying the rule →I backwards; however some were frustrated by not being allowed to use ∧I forwards (e.g. creating P∧Q by selecting P, selecting Q, and applying ∧I).

Feedback

On paper, errors in proofs were frequently undetected by students; and explorations of different routes to a proof were rarely deep. Lab assistants, lecture notes, fellow students, and tutors were relied upon both to validate the correctness of proofs and to provide hints as to how to proceed. The use of the lecture notes as a key authority was problematic for conjectures either that had a superficial similarity to conjectures in the notes or that did not appear in the notes. Textbooks were mostly not consulted. Meanwhile in Jape, students trusted the program not to make illegal steps; and students who overcame forward-fixated reasoning successfully used the fast feedback and the undo facility to explore how to proceed. When stuck, students tended to seek help from lab assistants rather than lecture notes, fellow students, or tutors; however this is likely to be related to the organisation of the sessions when the program was used. The ItL Jape manual was mostly not consulted. However, steps that created sudden enlargements in the size of the proof, or introduced unknowns, bifurcations or "inscope" were often treated as erroneous. This misdiagnosis was particularly acute in the Disjunction and Quantifier topics. Checking that it would be possible to prove a later line from earlier lines (by interpreting the logical connectors) was not a common strategy, in spite of much prior instruction in the semantics of formal logic. There is little evidence of attention to the justifications in deciding what rules to apply and where; and students' focus for attention generally seems to be either the most complex line or the lines either side of the ellipsis. The ellipsis acted as an important visual cue of "work to be done" and provided a satisfying feeling of completion when the proof was finished.

Speed of interaction

On paper, proving was slowed down by the need to draw boxes, and it also seemed almost as if the untidiness created by box-drawing tended to diminish students' appreciation of their work. In addition, incorrectly positioned boxes often led students astray, even when their basic plan for constructing the proof was initially sound. In Jape, box-drawing was automated and many students said that they valued this aspect. However, knowing which lines the box should encompass is of course an important skill, and heavier users of the software were not noticeably more skilful in this respect when returning to proving on paper. Much of the interaction with the software could be carried out solely using the mouse, and the keyboard was not needed for many steps. Most students expressed their appreciation of this, however a few students suggested that they would prefer to type in successive lines for checking, rather than have Jape generate the lines in response to mouse clicks. Interviews and observations suggest that students tended to tackle a far greater number of conjectures in Jape than on paper. One disadvantage of the speed with which the software carried out rule applications was that it was sometimes difficult to grasp what had just happened; however, the undo and redo facilities were then often used to replay the step.

Attention to structural aspects of proofs

On paper, little attention was paid to the structure of conjectures, or to the similarities between proofs. In Jape too, only a few students appeared to notice these aspects. Indeed, sometimes - both on paper and in Jape - students had more difficulty with the second attempt at a repeated proof segment than the first. In Jape, there were examples of students attempting to over-generalise from one rule to another, particularly from →I backwards. Issues of how to make the most efficient proof did not arise. However, it was noticeable that at various points in some complex proofs, that students suddenly became confident (not always reliably) about the success or otherwise of the particular approach taken. Students also appeared to have a heuristic order of precedence of rules: →I backward, →E forward, ∨E forward, ∧I backward, ∨I backward, ¬I backward, ∧E forward, ¬E forward, ¬E backward, ∀I backward, ∃E forward, ∃I backward, ∀E forward.

Perceptions of difficulty

On paper and in Jape, students found some rules harder than other rules. Implication and Conjunction tended to be seen as the easiest topics; Disjunction was next; Negation and Quantifiers were held in about equal dread. This order matches the order in which the rules were introduced in the lectures, the order in which the rules were practised on paper, and the order in which the conjectures were presented in ItL Jape. It was also observed that conjectures without premises seem to be viewed as potentially harder a priori.

Perceptions of proofs

Analysis of the discourse used in interviews and proof episodes suggests that the typical paper-based perception of a proof as a written, linear sequence of logical formulae contrasts with a possible perception in Jape that a proof is a set of simplifications of the conclusion and premises. Rules feature as technical warrants for lines when during paper proofs, but they are "applied" in Jape to generate lines and justifications automatically. One effect of this difference in perception might be whether students focus on complicated formulae or on missing justifications when deciding how to proceed.

Recall

There is clear evidence of some students struggling to make progress in their second Jape session, when they started with the conjectures on which they had been working at the end of the first session the previous week. Progress was only made when they returned to proving a few earlier conjectures. There is insufficient evidence about this aspect for work on paper. However, there are also indications of the same students being able to make fast progress four months after working on any proofs, when they worked through the conjectures in ItL Jape from the beginning.

Summary of Jape's advantages

In conclusion, therefore, the main advantages of ItL Jape for many students seem to be that it allowed them to consider many more examples than would be possible on paper, it encouraged experimentation with different routes to a proof, and it challenged inaccurate and forward-fixated reasoning.

Modelling the learning mechanisms

Proof strategies

Modelling students' knowledge by means of conjectured proof strategies helps in understanding the reasons for students' success in learning from Jape, and for their failure to learn from feedback in particular situations. These strategies vary in how they are combined and in how they are articulated. Rule-specific strategies have been identified that enable students to decide what rule should be applied in given situations, to implement that rule, and to predict the effects of the rule. Global strategies have been identified that help students to plan how they will attempt a proof, and to debug that plan.

Prior knowledge

Four groups of users were identified, based on their prior knowledge of the rules. Group 1 users know the name of the rule they want to apply but are not necessarily aware of what the precise effects of the rule might be; Group 2 users know how they want the transformed proof to look, but are less sure about the name of the rule that achieves this transformation; Group 3 users have a partial understanding of the rules and are trying to understand the rules from the output of the program; and Group 4 users have never met the rules before and so are not target Jape users.

How Group 1 students learn from Jape

It has been argued that Group 1 students can use Jape to improve their already functional rule-choice and progress-assessment strategies in increasingly difficult proofs, provided they quickly appreciate that structures are generated by rules.

How Group 2 students learn from Jape
Group 2 students, meanwhile, even though they have good progress-assessment strategies (they know what they expect to see) may find it difficult to improve their step-choice strategies using Jape because their step-name strategies are undeveloped. Their typical fallback global strategy «When all else fails, assume something» is particularly unhelpful in Jape, as is the break-down/build-up strategy suggested by some tutors - «Break down complex premises into components using elimination rules, and then build up the components into conclusions using introduction rules». Group 2 students therefore gain the least from the software.

How Group 3 students learn from Jape
Finally, Group 3 students *may* be able to use Jape to improve on their embryonic rule-choice and progress-assessment strategies, but whether they do depends crucially on these initial strategies. It may be enough at the start to have a symbol-matching rule-choice strategy: something akin to «Click on anything complex you haven't clicked on yet, find a rule that matches the principle connective, and undo if the result does not look like progress». But the break-down/break-up strategy suggested by some tutors - «Break down hypotheses using elimination rules, and break up conclusions using introduction rules» - would appear to be too unwieldy to be readily usable by Group 3 students. It is particularly important for these students to be systematic, to avoid any additional complexity, and to recapitulate soon afterwards in another form what they have learned. In the short-term, these students have the potential to gain most from Jape.

What students learn
In conclusion, then, as regards improving proof strategies, Jape would appear to be more supportive of those who are willing to explore than of those who want to reproduce paper proofs. However, none of the students are likely to develop new written rule-implementation or justification strategies when using Jape. Nor are they likely to develop a semantic-checking strategy such as «When reasoning backwards, check if the lines produced are impossible to prove from the premises; when reasoning forwards, check if the lines produced are unhelpful in obtaining the conclusion». Nor are they likely to develop a theorem-application strategy such as «If stuck on a conjecture with no premises, try to think of a theorem that might help with forward reasoning».

Explaining the differences in proving behaviour on paper and in ItL Jape

Forward-fixated reasoning
Forward-fixated reasoning was challenged by Jape because it constrained the strategies available; «make an assumption» was not possible, but «apply →I backwards» was; «assume the complement and aim to derive a contradiction» was not possible, but «¬I backwards» was. Consequently, most students soon discovered that the only way to generate structure was by applying rules; however, those who failed to realise this - and Group 2 students would be most at risk here - struggled to make much progress. That conjectures without premises were considered harder may be symptomatic, because no forward moves would appear possible. Nevertheless, just being forced to carry out a backwards step rather than a forwards step does not in itself reinforce the strategy of reasoning backwards in certain cases as being superior to reasoning forwards - as demonstrated by the frustration with ∧I forwards. The transfer of the strategy to paper depends on students appreciating its superiority.

Feedback
The poor progress-assessment strategies exhibited by many students suggest that they are Group 3 students, in spite of the prior instruction. Consequently, features that contributed to the systematic testing of embryonic rule-choice and progress-assessment strategies would be appreciated - such as the similarity of the display to the written version, the familiar point-and-click interface, the accuracy, the fast feedback, the automatic box-drawing, the undo facility, the gradual increase in the complexity of proofs, the indication of progress via the list of completed conjectures, and the ellipsis. Features that were irrelevant to this testing would be ignored - such as justifications, structural aspects of proofs, efficiency and semantic considerations. Features that inhibited this testing would be problematic - such as troublesome inputs (misclicking, ambiguous labels, an uncomfortable menu structure, parameters, text-selection, multiple possible conclusions), complex outputs (dramatic changes in proof size, bifurcations, boxes, variables), and unfamiliar outputs (unknowns, "inscope", opaque dialogue messages).
Although it is plausible that Group 3 students are less likely than Group 1 or Group 2 students to be able to distinguish a productive step from an unproductive step, evidence has been obtained of a Group 2 forward-reasoner and a Group 3 student with forward-reasoning tendencies working together on paper and then in Jape. The Group 2 forward-reasoner made little progress in her proof strategies, whereas the Group 3 student became a Group 1 student at the end of an hour, showing flexibility as to rule direction.

Speed of interaction
Students could tackle many more proofs in Jape than on paper because of the speed with which proofs could be constructed using the mouse, because illegal moves were challenged instantly (eliminating the need for repeated checking of one's work for legal rule applications), because the ellipsis constituted a visual cue to focus attention, and because the messy task of drawing the proof was handled automatically.

Attention to structural aspects of proofs
Students' sudden feelings of confidence about the success of the particular approach taken can be explained as suspicions that the structure of the proof was becoming like or unlike structures featuring in the idealised combination of strategies being used in the plan.

Perceptions of difficulty
The features that inhibit strategy testing go some way to explain the relative perceived difficulty of the topics. The Quantifiers topic involves variables (and not just propositions), is prone to incomplete steps (and hence bifurcations, unknowns, and "inscope"), has unknowns that represent variables, and has an order of rules for handling variables (∀I and ∃E before ∃I and ∀E). Avoiding incomplete steps involves passing parameters, both propositions and variables (selected and text-selected). It might also be easy for dyslexics to confuse "∃" with "E" at first glance. Disjunction has bifurcations (∨E), ambiguous labels (∨I), and multiple possible conclusions. Although difficulty with Negation is partly explained by the lack of organised Jape sessions devoted to it, its difficulty is also likely to be related to the fact that the symbol-matching rule-choice strategy cannot suggest the ¬E¬I∧I combination. Progress-assessment in this case also often seems to be supported by semantic checking, which is likely to have been ignored up until this point.

Perceptions of proofs
It seems plausible that Group 1 and Group 3 students using Jape are likely to end up seeing a proof as a set of simplifications; whereas Group 2 students are more likely to emphasise the linear, justified sequence.

Recall
Group 3 students have to develop a large number of rule-specific strategies in a relatively short time. It is therefore unsurprising that they would have difficulty in remembering them a week later. The conjectures that they had found difficult even when the strategies were freshly formed were consequently a poor place to start subsequent work. Recapitulation is vital for these students until they have found ways to remember the strategies. Students appeared to attempt a variety of means to cope with the memory demands of these strategies; for example, reviewing already completed proofs, tackling particular difficult proofs again to check that the method can be reproduced, tackling easier proofs from the previous session to recapitulate the strategies, and replaying steps that had a dramatic effect on the proof. One unsuccessful attempt was to overgeneralise strategies between rules.

Enhancing strategy development

Forward-fixated reasoning
In the next version of ItL Jape, hints as to which steps create assumptions, introduce variables, and use variables are indicated on the menus; this may help to dislodge forward-fixated reasoning and create greater awareness of the structure of the rules.

Recall
Recall might be assisted by icons, mnemonics, sounds or animations that suggest the structure or actions of the different rules without bypassing the names of the rules. Showing the ∨E boxes horizontally rather than vertically has been suggested as another way of emphasising the rule's structure.

Rule-implementation and justification strategies
The automatic drawing of boxes and the automatic insertion of justifications left the students free to focus on debugging their rule-choice and progress-assessment strategies; but rule-implementation and justification strategies were unaffected. One way of enhancing the educational value of the tool could be to provide novices with options to turn off automatic boxes and justifications once the basic uses of the rules have been understood, so that the user would have to draw boxes with the mouse and enter justifications with the keyboard, and so develop rule-implementation and justification strategies.

Semantic checking
Because, as already mentioned, students rarely used a strategy of semantic checking, they sometimes spent virtually all their time on a proof having reached a situation in which by interpreting the logical connectors it is clear that it is impossible to prove later lines from earlier lines. A human tutor would likely spot such impossible situations, and would perhaps urge the student to check, or to undo steps. Although it is not possible to trap all erroneous steps, it

is possible that the software could recognise the impossible situation and then alert the user to it if he or she failed to notice it after a certain length of time, after a certain number of steps, or after the proof had reached a certain size. It is possible that students would initially use such an alert purely as an error indicator, but that they would in time begin to check for themselves.

Proof efficiency
Could Jape promote efficiency strategies? For example, there are proofs where applying the rules in a different order can result in a shorter proof; and seeking theorems to apply can also make some proofs easier. One suggestion is that, once the student has completed a proof, the software advises the student if such strategies are possible, and encourages a re-attempt.

Terminology
The next version of ItL Jape allows students to obtain assistance on important terminology, such as "hypothesis", "conclusion" and "unproved conclusion". This should help to reduce dependency on external sources of help.

Hypothesis/conclusion selection
In the light of this research, the next version of ItL Jape has a feature by which the selection box for a hypothesis is open underneath, and that for a conclusion is open on top, the idea being to reinforce the distinction between hypotheses and conclusions.

Further refinements to the interface

Comprehensibility of dialogue messages
Because most of the dialogue messages were not comprehensible to the students (they were not intended to be), the students tended to ignore the text and treat the message merely as an indication that they had attempted an illegal step. Those messages that *were* informative were therefore not read properly; and this led to particular difficulties with situations when the user had to resolve an ambiguity about the conclusion towards which a rule application is intended to work. As a result of this finding, the next version of ItL Jape has been given substantially improved dialogue messages: they have been carefully tailored to specific common problem situations; they indicate more clearly than before the nature of the difficulty (distinguishing illegal rule applications from incompletely-specified rule applications); they indicate what action the user attempted (thus helping the detection of misclicks); and they suggest possible resolutions.

Display of dialogue messages
There were two further difficulties with the dialogue boxes. Firstly, they tended to distract students' attention away from the body of the proof, because they have to be dismissed with a click. Secondly, students were unable to continue working on the proof until the dialogues were dismissed, which meant that the information they contained was lost while students attempted any suggested actions. One potential resolution of these difficulties that could be explored is the use of an advice panel instead of a dialogue box. Such an advice panel would be displayed permanently on the screen; and so the distraction aspect would be reduced because no clicking would be required, and the information would remain on the screen while students attempted suggested actions. Such a feature could also have the advantage of letting students peruse a "history" of attempts, thus minimising repetitive failed attempts; and if successful rule applications were also displayable, perhaps students might also be able to use the history to detect patterns in the sequence of applications. This quieter form of messaging would be more in keeping with the interface philosophy than dialogue boxes, although the feature would perhaps clutter the screen, so expert users may prefer to have an option that allows them to turn off the advice panel and replace messages with a beep.

Incomplete steps
ItL Jape's treatment of incompletely-specified steps was a source of a number of difficulties. Two key interface strategies refer to how to use ItL Jape to apply the rules: «Before applying a forward rule, click the main hypothesis» and «Before applying a backward rule, click the conclusion». These two strategies could be summarised in one: «Select what you want to simplify». However, in some of the simple situations met early in ItL Jape, it is possible to apply a rule successfully without having selected a line (the program works it out) and this success may have counterproductive in that students later failed to recognise that some unexpected outcomes were the consequences of not grasping this interface strategy. For example, when students failed to click a line (or clicked the wrong line) before applying certain rules, Jape interpreted the rule application as an *incomplete backward step*; whereas the students had in fact intended a *complete forward step*. There were two main difficulties with this. Firstly, the direction may have been opposite to the one intended; and students typically failed to realise that this was because they should have clicked particular lines. Secondly, unknowns may have been created; and because many students did not know what the artefacts were for, they misinterpreted the arrival of unknowns as indicative of an illegal step (rather than of an incomplete step). These students would then typically be misled into

trying a wrong rule, an outcome that could undermine students' tentative theories about the structure of the rule and their strategies for using it. It would appear plausible that placeholder unknowns are most useful for more expert Group 1 users as tools for exploring possible avenues, but that they are intimidating for novice Group 2 and Group 3 students. In the light of these findings, the next version of ItL Jape has implemented "training wheels" - a feature that *requires* students to specify the formula to be simplified, and puts on the menus only those steps that novice students are likely to use (thus removing many of the opportunities for incomplete steps). None of the steps that are missing from the menus were intentionally chosen by any novice students observed. Ideally perhaps the training wheels could be removable at a student's request.

Misclicking
The misclicking of items on the rules menu, while not frequent, tended to cause confusion when students were *unaware* that they had misclicked. Moreover, when students were *unsure* as to whether they had misclicked, they would have to undo and reapply steps in order to check; and errors in reapplication compounded the confusion. Removing the dash in the description of each rule may assist readability and so reduce misclicking; another idea mentioned below that might minimise misclicking is reorganising the rules menu. A number of changes might enhance the detection of misclicks: encouraging the use of the "tear-off menu" feature (which would allow students to see clearly the rule just applied), indicating on the "undo" menu item or tool icon the name of the step that would be undone (as in Microsoft Word), indicating the action attempted in dialogue messages, providing a history of rule applications, providing a tree display as an alternative to the box display, and highlighting changes to the proof in a different colour.

Inefficient mouse movement
One commonly observed phenomenon is the rapid yet inefficient traversal of the rules menu by the mouse pointer. An analysis in terms of strategies by the different user groups suggests that if this indicates difficulty in finding the rules then dividing the rules menu into forward and backward rules rather than introduction and elimination rules might help. This change would perhaps fit better with some students' decision-making processes, and would fix the problem described earlier in which the user was surprised at the direction a rule would take. The change has been made in the next version of ItL Jape, although it may not suit all students and so might usefully made an option.

Double-clicking simplification
Jape could offer a feature that enabled students to simplify formulae by double-clicking rather than selecting a rule. An analysis of the effects of this feature in term of proof strategies suggests that there are advantages for Group 1 students who are already competent with the rules. However, although double-clicking simplification may help forward-fixated Group 2 and Group 3 students to reason more flexibly, they might not gain competence that transfers to paper.

Forward-reasoning restrictions
Group 2 students would appear to have the least productive experience of Jape. If these students could be identified accurately, one way of making the program more productive for them might be to turn off at least some of the forward-reasoning restrictions that inhibit sub-optimal steps but that consequently jar with these users. However, if this were done, the program would need to show context-related hints in order to encourage flexible reasoning.

Speed of rule application
Some rule applications surprised students by the sudden contraction or expansion of the proof. A human tutor would take such steps more slowly; perhaps there could be an option to animate changes, or to show changes in a different colour.

Additional interface principles
Given that students were using an interface strategy akin to «Select what you want to simplify», some difficulties can be predicted in situations where further principles are required. Three examples are given below: understanding the (L) and (R) labels; dealing with unknowns; and indicating the conclusion towards which a rule application is intended to work. However, such interface difficulties could be attributable to lack of experience with using ItL Jape - most students spent less than 90 minutes using the software and the more experienced users found fewer difficulties.

Left-right confusion
Some students found the labels (L) and (R) ambiguous, in that they were unsure whether, for example, it is ∧E(L) or ∧E(R) that concludes P from P∧Q: Is it by using ∧E(R) to *eliminate the term on the right*; or is it by using ∧E(L) to *eliminate the ∧ and leave the term on the left*? As a result of this finding, the next version of ItL Jape indicates on the rules menu which side is discarded for the ∨I backward steps and the ∧E forward steps, and which side has an unknown term for ∨I forward. Another approach would be to allow users to indicate - after the selection of simply "∧E", say - which term they wished to see appear (or to see justified) in response to a dialog message.

Dealing with unknowns
Students had difficulties with dealing with unknowns, difficulties related in particular to text-selection, unification and parameters. When unifying unknowns and passing parameters, many students were initially unaware of the distinction between *selection* and *text-selection*; they were then confused about when to select and when to text-select; they found carrying out text-selection physically difficult, particularly multiple text-selections; and cancelling a text-selection was not always successful. It would appear that the introduction they had been given to Jape and the contextual online help were not enough to help them sort out the details without additional help. However, once demonstrated, these skills were rapidly acquired with practice. Over and above their difficulties in recognising unknowns as unknowns, and in using text-selection, students often had difficulty in working out or remembering that to provide a reference for an unknown they had to use either the "unify" command or the "hyp" rule (students repeatedly mentioned the hyp rule as a feature they did not understand). This may have added to the perceived difficulty of the Negation topic. Alternative mechanisms for unification could be explored, such as allowing references to be selected from a pop-up menu or created in a dedicated text-selection mode. A particularly common error in unifying unknowns was failing to text-select the underscore in front of the unknown. It was also apparent that some students did not know or had forgotten that parameters could be passed to rules, while others were not sure *what* parameters could be passed to each rule.

Conclusion ambiguity
Although the improved dialogue messages in the new version of ItL Jape may help students to deal with the situation in which there is more than one conclusion towards which a rule application might work, the strategy «Select what you want to simplify» is again compromised. An alternative mechanism that is perhaps worth exploring is to allow the selection of the conclusion formula *after* the rule application if the user desired.

Colour selections
When a line is selected, a box appears around the line. For the novice, this extra box could be distracting. However, Jape can make this box a different colour to the black boxes surrounding groups of lines, and this feature has been introduced in the next version of ItL Jape.

Done button
Students were not always aware that completed proofs needed to be registered. Several students suggested a big "Done" or "QED" button as a way of finishing off a proof, or that Jape itself acknowledges that a proof is complete.

Window jumping
Some minor inconvenience was caused by the proof window suddenly partly disappearing off-screen and students struggled to find a way to re-centre it. This movement made it harder to students to track changes in the proof.

Prerequisite knowledge to use the program

The demands of the interface can easily be listed, three key points being how to apply a rule, how to assign values to unknowns and how to pass parameters. The demands of the logic are not so clear; but elemental rule-choice strategies and progress-assessment strategies are conjectured prerequisites.

Methodology

The research value of a logging mechanism has been clearly demonstrated here, when combined with qualitative approaches to understanding learning situations. Videos of students proving - on paper, then using Jape, and then on paper again - have made the greatest impact on developing enhancements to the program. The notion of proof strategy has also shifted the problem from "How can we teach students to prove?" to "What are the circumstances under which students find it easiest to refine proof strategies?".

Further research

Interface changes

In the light of this research, a number of changes to the program have been made or mooted. Assessing whether these changes enhance students’ experience with proving using the software would assist in the corroboration of this analysis. For example, the next version of ItL Jape offers tailored dialogue messages, an annotated and reorganised rules menu, and an option that requires users to specify parameters for most rules. These changes could be usefully evaluated using logfiles and questionnaires. Further interface variants might best be evaluated using video and interviews: an advice panel, optional automated box-drawing and justifications, iconic representations of rules, double-clicking simplification, permitted forward-reasoning, unification mechanisms, and the use of proof trees as an additional, alternative representation.

Transfer of knowledge

It would be interesting to find out the usefulness of taking a logic course as regards the overall Computer Science degree. When they come to planning software, for example, do students somehow draw on their experiences of studying logic, by making use of notions of “rigour” or “completeness” or “logical structure”?

Proof system

Some of the results obtained relate to this particular implementation of the natural deduction system. Further work could be done in order to explore which results hold for other logics, and whether there are more “natural” proof systems for introducing formal reasoning than this one - changing the rules for Negation could make the system slightly more intuitive.

Students’ proof behaviour

More detailed logfiles would enable an exploration of students’ heuristic order of precedence of rules. Such logfiles could also help investigation of whether some students find it harder than others to break out of a symbol-matching rule-choice strategy. In addition, it was found here that students had difficulty in remembering the many rule-specific strategies learned; further research could investigate what might help recall, including longer exposure to Jape, firming up conjectured prerequisite knowledge, building recapitulation into each session, and encouraging students to reflect on why particular conjectures should be believed and on the reasons for a proof not working.