The Evaluation of a Computer Program for Learning Logic: The Role of Students' Formal Reasoning Strategies in Visualising Proofs

James Aczel
The Institute of Educational Technology, The Open University,
Milton Keynes, MK7 6AA, UK, Tel: + 44 1908 652953, email: j.c.aczel@open.ac.uk

Abstract

This report describes empirical research into the effectiveness of a program called Jape in supporting the learning of logic. By comparing undergraduates' proving behaviours on paper and in Jape, it is hoped to increase understanding of the features of such tools that support the learning of formal reasoning for software development.

Jape allows users to manipulate proofs using a mouse. An educator can control which logic Jape uses, how proofs are presented, and what actions the user can take. The implementation studied here - ItL Jape - is pre-loaded with propositional and predicate logic, with Natural Deduction rules and with Fitch boxes. When users apply a rule to a line of the proof, the software shows the effect on the proof.

Three studies are described. In the first study, four students were videotaped as they used the program to assist them in constructing proofs as part of their course. In the second study, data was collected on the whole cohort: students' backgrounds, usage of Jape, and success in the course. In the third study, ten students were videotaped using the program in task-based interviews.

A quantitative analysis of tests, surveys and logfiles suggests that students' backgrounds appeared 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 was a significant factor in course performance, even when significant background variables such as gender, degree course, and prior programming experience are taken into account.

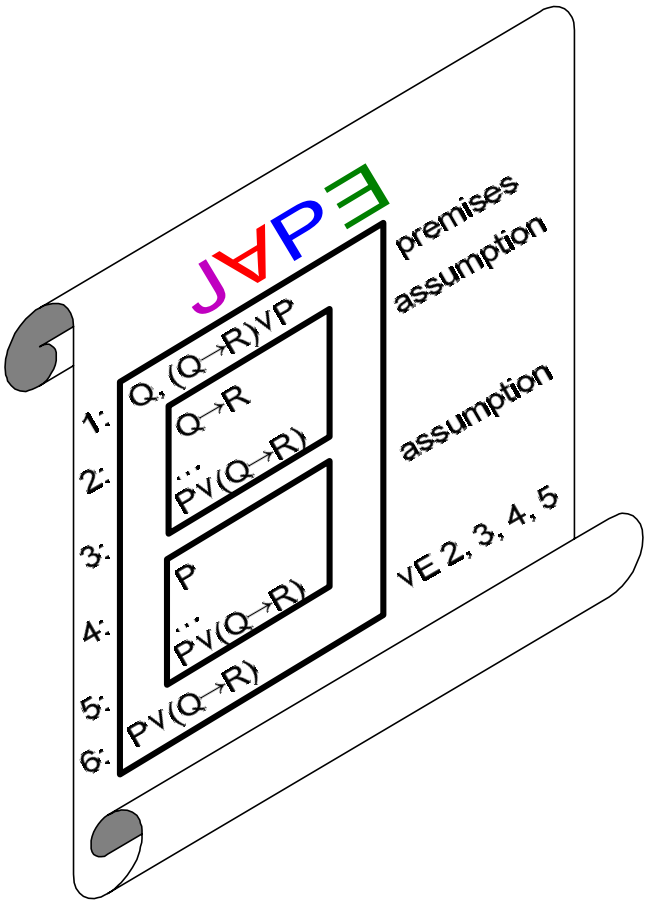
Evidence from observation and interviews suggests that the main advantages of ItL Jape for many students are that it allowed them to consider many more examples than would be possible on paper, it encouraged experimentation with different routes to a proof, and it challenged inaccurate and forward-fixated reasoning.

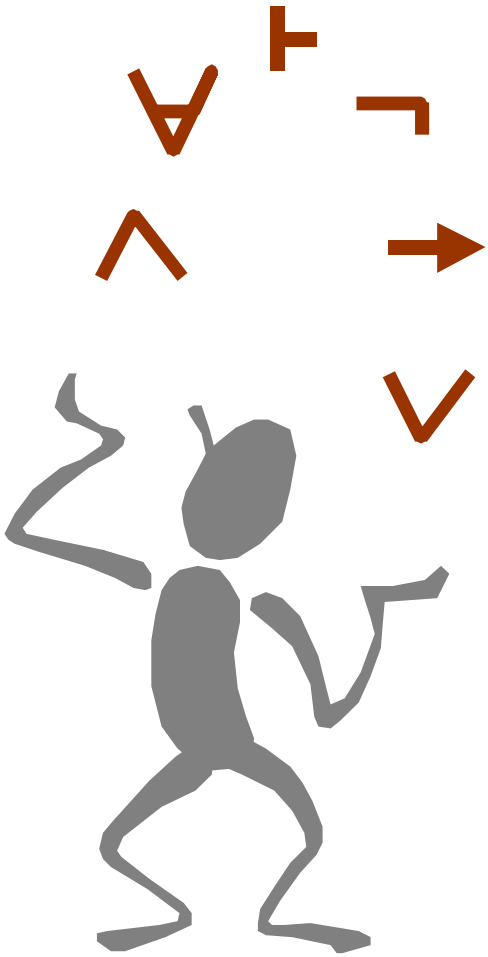
A model of students' knowledge in terms of conjectured proof strategies is developed. Rule-specific strategies that enable students to decide what rule should be applied in given situations are distinguished from global strategies that help students to make and debug plans for tackling the proof. Students are categorised by their prior knowledge of the rules. This model provides insights into the reasons for success and failure in learning from Jape, explains the differences in proving behaviour on paper and in ItL Jape, and enables predictions about interface refinements that would enhance learning.

This research was funded by the Engineering and Physical Sciences Research Council, as part of the project "Visualisation in the Software Development Process"

Although this particular report has only one author, the research has been carried out in collaboration with Pat Fung (IET), Richard Bornat (Queen Mary & Westfield College), Martin Oliver (University of North London), Tim O'Shea (Birkbeck), and Bernard Sufrin (University of Oxford). Their generous contributions to this work are gratefully acknowledged. The main project papers listed in the references are all jointly authored.

Contents


∀∃P∃
premises
assumption
assumption
∨E 2, 3, 4, 5
1:
2:
3:
4:
5:
6:
Q, (Q→R)∨P
Q→R
...
P∨(Q→R)
P
...
P∨(Q→R)
P∨(Q→R)



1. Introduction

There is evidence that many students can find formal reasoning difficult (Fung et al, 1993, 1994). Yet it is also clear that there are programs - such as Tarski's World (Barwise & Etchemendy, 1992) - that are considered by students to be useful and enjoyable for learning the syntax and semantics of first order logic (Fung & O'Shea, 1992; Fung et al, 1996). Nevertheless, the notion of *proof* is notoriously not well appreciated by students at high school or university.

Can students be helped by software tools to understand the nature of formal proof? *How* exactly can students use a software tool to learn to construct proofs? What aspects of the interface and functionality might be vital for non-superficial understanding?

The program Jape (Bornat & Sufrin, 1996) allows interactive, step-by-step construction of proofs for a variety of logics. It allows a teacher some degree of control over the rules that can be used, the display of the proof on-screen, and the effects of mouse clicks. By researching students' experiences with Jape and by trying to discern the cognitive processes at work when students work on proofs, it is hoped to increase understanding of the effectiveness of such tools in supporting the learning of formal reasoning for software development. The implementation of Jape used in this research shares some similarities with MacLogic (Dyckhoff, 1987) and the Carnegie Mellon University Proof Tutor (Scheines & Seig, 1993), but the style of the graphical interface is innovative in its "quietness".

This report describes some empirical research into Jape's effectiveness in supporting an undergraduate course in first-order logic using natural deduction. The research aims to explore the benefits Jape offers students, the cognitive processes at work, the differences between student groups, and how the program might be improved. In doing this, it is hoped to increase understanding of the features of such tools that support the learning of formal reasoning for software engineering. The role of visualisation in such support is of particular interest.

After an outline of the specific implementation of the program evaluated in the research, three overlapping studies are reported. In the first study - "The Observational Study" - volunteer students were videotaped as they used the program to assist them in constructing proofs.

The second study - "The Measurement Study" - involved the collection of data from the whole cohort on background variables, Jape usage, and course outcomes. Written tests, surveys, and logging of program usage were the main data collection methods for this study.

The third study - "The Reflection Study" - again involved the videotaping of students using the program, but this time the volunteers were given specially-designed exercises in a task-based interview setting, rather than being observed in a naturalistic setting.

2. About the program Jape

Jape is a program that allows users to manipulate proofs using a mouse. An educator can control which logic Jape uses, the presentation of the proofs for that logic, and the effects of user actions. Because of this generality, it is argued in this section that, for the purposes of this research, a particular implementation of Jape must be studied. This implementation ("ItL Jape") is described.

2.1 The need for a particular implementation

The program Jape can operate with a variety of logics, by taking a description of a particular logic as a system of inference rules. It has been applied, for example, to predicate calculus, Hindley-Milner type assignment, axiomatic set theory and a functional programming logic. How proofs are presented is under the control of the person who codes the logic. Jape also has a tactic language in which actions may be bound to mouse clicks, menu items and keystrokes, and so the proof tree in any logic can be directly manipulated simply by clicking a formula with the mouse.

Among recent attempts to evaluate the educational potential of logic software, Kadoda (1997) compared features available across a wide range of theorem-provers (some of which might be used in educational settings), using a standard questionnaire given to program users and developers, and based on the vocabulary of "cognitive dimensions" (Green, 1989). Van Ditmarsch (1998), meanwhile, compared five natural deduction proof assistants using issues such as how proofs are displayed, bias towards either forward or backward reasoning, and the availability of help.

However, in order to address the question of how precisely Jape might assist learning, it is clearly necessary to examine students' experiences with logic at a level of detail that enables conclusions to be drawn from specific interactions with the software. Hence, a particular implementation of Jape is required.

2.2 ItL Jape

The implementation of Jape investigated in this research is used on an introductory course in propositional and predicate logic for first-year computer science undergraduates. It was pre-loaded with the "natural deduction" style of reasoning used in the course; it was set so as to present users with a sequence of conjectures to prove; and it was also configured to require the user to indicate, step-by-step, which rule is to be applied to which line. In other words, it has been configured - with learners in mind - to act more like a logic calculator than a theorem-prover.

This implementation of Jape is called for the purposes of this research "ItL Jape" in order to highlight that by focusing on a specific implementation of the software, this research is mostly unable to evaluate Jape's flexibility with respect to logic system or with respect to interface design. On the other hand, by examining in detail students' interactions with a specific implementation, it may be possible to indicate something of the potential strengths and pitfalls that might apply to other proof assistants.

The course lecturer decided not to provide ItL Jape with direct manipulation rules (so, for example, double-clicking on a formula would apply the most "obvious" rule) because it was suspected that novice logicians would learn more about natural deduction if they had to choose the rule for themselves. But Jape could obviously be configured with different users in mind - more experienced logicians might want different aspects automated. Note also that Fitch boxes (Fitch, 1952) were chosen for displaying proofs, rather than proof trees.

The rules of the logic have been put are on a menu, and are listed in the appendix to this report.

As an example of its operation, consider the conjecture Q→R ⊢ (R∨Q)→(P∨R). ItL Jape would initially display:

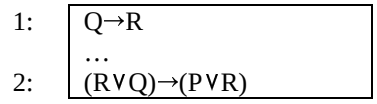


Figure 1: How Jape shows the conjecture Q→R ⊢ (R∨Q)→(P∨R)

Clicking line 2 and selecting "→-I" from the rules menu changes the display:

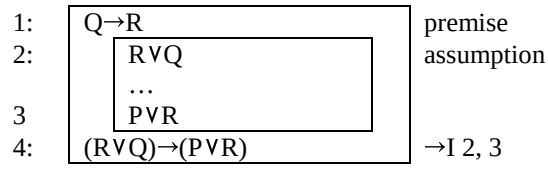


Figure 2: Q→R ⊢ (R∨Q)→(P∨R) **after →I is applied to line 2**

In effect, the "implication-introduction rule" has been applied.

A student might proceed from here by using ∨I(L) ("or-introduction left") on line 3. This would lead ultimately to a dead-end. Applying ∨E ("or-elimination") to line 2 is more useful:

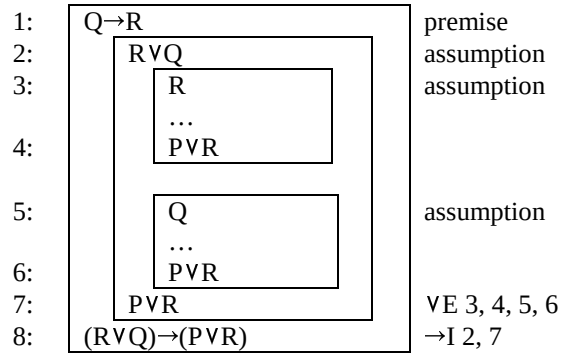


Figure 3: Q→R ⊢ (R∨Q)→(P∨R) **after →I is applied to line 2 and ∨E is applied to line 2**

And so on, until the proof is complete. The ellipsis symbol "…" then disappears.

ItL Jape is the implementation considered by van Ditmarsch (1998) to be the most visually "appealing" of the five proof assistants he scrutinised, although he notes that "There is no proof help, apart from 'debugging' help: helpful warning messages when wrongly applying rules." and "It is not possible to submit entire proofs. It is not even possible to make an incorrect proof step, as rule execution is automatic given a formula and a rule. This makes Jape less fit for teaching natural deduction to 'absolute beginners'.".

2.3 The conjectures contained in ItL Jape

The conjectures pre-loaded in ItL Jape are given in the appendix. There are 70 conjectures in total, presented in a long list in the order given in the appendix, but without the identifying numbers and topics that have been ascribed for the purposes of this analysis.

There are 11 conjectures (C1-C11) for which the only operator is →: this operator is syntactically referred to as "arrow" or "the implication symbol", semantically referred to as "implies". These conjectures are described here as constituting the "Implication" topic. There are 5 conjectures (C12-C16) for which the only operator, apart from →, is ∧ ("and" or "the conjunction symbol"). These conjectures are described here as constituting the "Conjunction" topic. There are 13 conjectures (C17-C29) for which the only operator, apart from → or ∧, is ∨ ("or" or "the disjunction symbol"). These conjectures are described here as constituting the "Disjunction" topic. There are 18 conjectures (C30-C47) for which the only operator, apart from the 3 introduced so far is ¬ ("not" or "the negation symbol") *or* for which it is intended that students prove them using the negation rules. These conjectures are described here as constituting the "Negation" topic. There are 18 conjectures (C48-C65) which include the "quantifier" symbols ∀ ("for all", "for every" or "the universal quantifier") and ∃ ("there exists", "there is" or "the existential quantifier"). These conjectures are described here as constituting the "Quantifiers" topic. Finally, there

are 5 conjectures (C66-C70) which are false and are labelled on-screen with the word “NOT” following them. These conjectures are described here as constituting the “False” conjectures.

It is also possible for students to enter their own conjectures (the “User-Entered” conjectures). These are labelled here with a three-digit number, C100, C101, etc.

3. The Observational Study

The work of students was observed over the whole of a particular course in introductory logic. The bulk of this observation took place in weekly "laboratory workshops". This section reports on these observations, and in particular the videotaped data of a few individuals using Jape in the workshops.

3.1 The course

The course - "Introduction to Logic" - is a first course in logic aimed primarily at computer science students. The course aims to introduce students to fundamental notions in propositional and predicate logic, and to impart the skills of translating arguments into the formal languages, and of doing semantic and natural deduction proofs. The course is billed as being supported by two teaching programs: Tarski's World and Jape. The first is used to gain familiarity with the formal languages and the fundamental ideas of model theory; the second is used to gain familiarity with the proof theory and to develop the skill of finding proofs.

The justification given to the students for studying logic is that "Logic is fundamental to many areas of Computer Science. It is, for example, used in program specification and verification, artificial intelligence, databases, logic programming, and human-computer interaction.".

Around 170 students started the logic course. The teaching of the course took place through:

1. twice-weekly hour-long lectures to the whole cohort;
2. course notes, praised by the students to such an extent that many students considered studying the notes an adequate replacement for attending the lectures;
3. weekly "laboratory workshops", also known as "exercise classes", in which exercises based on the work taught in the previous week's lectures were undertaken in small groups with help available from 3-4 "lab assistants";
4. tutorial classes, involving each tutor teaching a small group of students; the format of the classes being largely at the discretion of the individual tutor.

Students were divided into four groups for the workshops, to provide more opportunities for questions and explanation. The main recommended text was Barwise & Etchemendy (1992); Hodges (1997) and Reeves & Clarke (1990) were also suggested as useful.

This study made use of a naturalistic evaluation approach (Guba & Lincoln, 1981), in that the students were observed working on course tasks over the whole of the logic course in order to understand how ItL Jape fitted into the learning context. The focus of the observational research was the weekly lab sessions, because they offered the opportunity to observe students actively engaged with the subject matter and to talk informally to them about their understandings and difficulties.

3.2 The workshops

The activities of the lab workshops are listed in Figure 4. Mostly students were encouraged to use pencil-and-paper, but Jape was used on 3 occasions.

Lab	Date	Activity
1	5 October 1998	Translation from English to propositional notation; and truth tables
2	12 October	Truth tables and semantic entailment
3	19 October	Natural Deduction: Implication, Conjunction, Disjunction
4	26 October	Natural Deduction: Negation
5	2 November	Jape workshop 1: Implication, Conjunction, Disjunction
6	9 November	Either revision class or Jape workshop 2: Negation
	13 November	Logic Test 1
7	16 November	Tarski's World workshop 1: propositional logic
8	23 November	Tarski's World workshop 2: predicate logic
9	30 November	Translation from English to predicate notation
10	7 December	Jape workshop: Quantifiers
11	14 December	Set theoretic and Natural Deduction proofs for predicate logic
	18 December	Logic Test 2
	17 May	Exam

Figure 4: The lab sessions and tests

3.3 Observations of students using Jape

In the rest of section 3, the conclusions from the Observational Study are presented. The evidence for these conclusions comes from three sources: comments from students during workshops, the video-tapes of four volunteers using ItL Jape during the three Jape workshops, and the interviews with these four students after each workshop. A summary of the video data available is shown in Figure 5.

Episode	Student(s)	Date	Conjectures attempted	Length of Episode	Notes
1	Kusi	2 Nov 98	1-18	47 mins	*First use of Jape. Fast progress.*
2	Lewis & Caroline	2 Nov 98	1-23	49 mins	*First use of Jape. Very fast progress.*
3	Kusi	9 Nov 98	18-22, 30-31	60 mins	*Second use of Jape. Slow progress.*
4	Lewis	9 Nov 98	23, 17, 19, 20	83 mins	*Second use of Jape. Little progress.*
5	Kusi & Yasmin	7 Dec 98	58-60	22 mins	*Third use of Jape. Little unassisted progress.*
6	Lewis	7 Dec 98	58-62	63 mins	*Third use of Jape. Little unassisted progress.*

Figure 5: Video data from the Observational Study

The first Jape session began with a half-hour presentation of Jape's features.

In this report, where reference is made to individual students, the names used are of course pseudonyms, and students can also be referenced by a number, for example "Kusi (S18) was videoed using Jape during both the Observational and Reflection Studies".

Of the volunteers, two provided most of the data - Kusi and Lewis. Kusi (S18) was registered for "Computer Science and Business Studies". She had mathematics A-level, a little programming experience, use of a computer a home for a variety of purposes and a financial motivation. She expected the logic course to be fairly easy but fairly interesting. She scored well on the maths test, slightly below average on Logic1, slightly above average on Logic2 and well above average on the exam (see section 3.10 for details of these measures). She spent about 2 hours using

Jape during the workshops (which counts as “high usage”) and proved 60% of the given conjectures (which counts as “medium usage”). However she also spent around 5 hours using Jape during the Reflection Study (see section 5).

Lewis (S108) was aged over 25, with a background in mathematics and computing (including programming). His motivations included following an interest and possible financial rewards. He expected the logic course to be fairly difficult but interesting. He scored low on the maths test, well above average on Logic1 and Logic2, but around average on the exam. His opinion of the course (just before Logic2) was that it had been very worthwhile, fairly easy, and interesting. He did not consider Jape helpful. He spent about 3½ hours using Jape (which counts as “high usage”), mostly on Implication, Conjunction and Disjunction. However, he proved only around 40% of the given conjectures (which counts as “medium progress”) and did not take part in the Reflection Study. In fact for some weeks before the exams he had been banned from using the computer rooms because of inappropriate activities.

3.4 Interface design issues

When observing students, often the most striking aspects brought to attention are interface issues.

From an HCI perspective, the authors of Jape aimed for a “quiet interface” that “shows its users just what they want to see [and] never surprises them with irrelevant detail” (p. 1). This was indeed a valuable aspect of the interface.

Moreover, they stressed that it should be *easy to start using the program*: it therefore adopts familiar gestures, commands and visual cues from the Apple Macintosh operating system, and it displays proofs that look just like those in the students’ courses It should also be *easy to explore proofs*: it therefore clearly shows the range of available actions, it always shows the effect of actions, and it allows a poor choice of action to be undone. Finally, its “use must reinforce and be reinforced by the habits of thought that the task requires.” (p. 3). They argued that “It is important that tools devised for the teaching of proof construction should have interfaces as quiet as possible, because the task itself is so intrinsically complicated that the tool must not intrude.” (p. 2).

The close similarity between the display of Jape and pencil-and-paper notation meant that students found little difficulty in interpreting the display after proof steps that were well understood were carried out as intended. One suggestion was that the two boxes generated by ∨E could be displayed side-by-side instead of vertically. This would have broken with the notation used in the lectures, and so was not an option. However, it would have emphasised the non-linear nature of these two boxes.

The use of the “undo” facility was widespread and often praised as a useful way to explore possible steps. There were objections that this reduced the exercise to mindless trial-and-error (see later), but most students recognised the value of being able to backtrack in this way. However, in saving work in progress, Jape saves the state of the proof, rather than the steps taken to get to that state. So it is not possible to undo saved proofs or “re-run” old proofs to see how they were accomplished.

Students were not always aware that completed proofs needed to be registered. For example, Kusi completed 7 or 8 proofs before realising that she needed to select “Done” from the Edit menu, to indicate that the proof was complete, and so was unaware that she had a large number of windows open. Although Lewis and Caroline remembered every time, the mouse lingered over the conjecture list on more than one occasion before moving to the Edit menu. Several students suggested a big “Done” button as a way of finishing off a proof, or that Jape itself acknowledges that a proof is complete.

Although students did not always remember to indicate that the proof was complete, this could be an indicator of the success of the ellipsis as a visual cue. It enables students’ attention to be subtly directed towards “work to be done” and gives a satisfying feeling of completion when it disappears after the proof is finished. This feeling of completion is absent from pencil-and-paper proving:

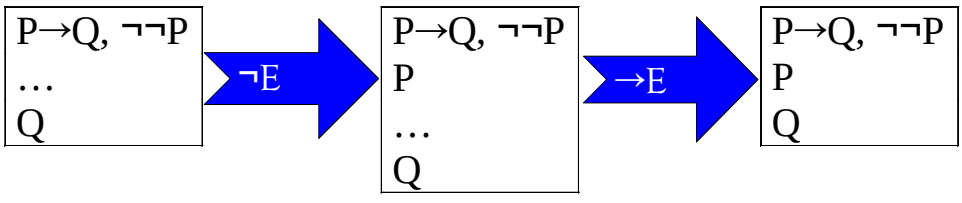


Figure 6: The feeling of closure when the ellipsis disappears at the end of a proof

Bifurcations, however, tended to raise doubts about whether the proof was any simpler:

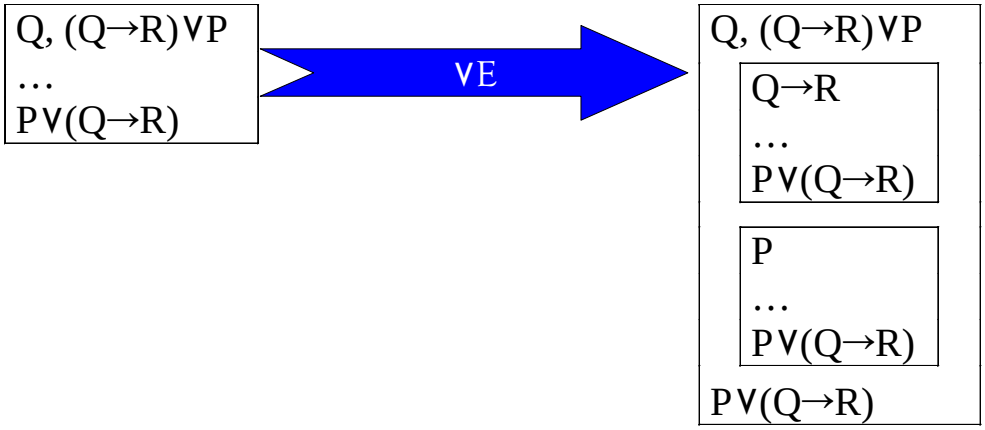


Figure 7: Bifurcations create anxiety

The fact that Jape drew the required boxes in the appropriate place was much appreciated. Many students commented on how time-consuming this is using pencil-and-paper; and students with an eye for tidiness work found the problem of the unpredictability of the box size was no longer an issue under Jape. 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.

Another timesaving feature was that Jape would automatically fill in the justifications for lines once a step had been carried out. Again, this allowed attention to be devoted to the sequence of steps rather than the formal output.

Much of the interaction could be carried out solely using the mouse. The keyboard was not needed for many steps. Most students expressed their appreciation of this, however Rachel (S72) suggested that she would prefer to type in successive lines, much as she would write them on paper, so that her proofs could be checked by Jape, rather than have Jape generate the lines in response to simple clicks.

Many students were unaware of the need for "text-selection" when passing a parameter (indeed, some did not know or had forgotten that parameters could be passed to rules), and many found carrying out text-selection physically difficult, although this was largely overcome with practice.

Lack of knowledge of the need for parameters had serious repercussions for students understanding. When steps were carried out without required parameters, Jape used "unknowns" such as "_A", "_B", "_B1", and "_B2" to allow the step to represented in any case. For expert users, these "placeholder" unknowns are extremely useful tools for exploring possible avenues (although some experts say that they prefer not to have to label what they don't know). However, these unknowns tended not to be recognised by novice students as unknowns, but to be taken as an indication of an *incorrect* step, as opposed to an *incomplete* step. For example, Kusi said, "When _A and _B came up you knew were on the wrong track" and later "I thought when you had A's you were doing it wrong".

Students would then tend to resort to trying other rules, or clicking other lines. This, it could be conjectured, tended to undermine their tentative hypotheses about how the rule could be applied, and to diminish their faith in their ability to select the correct rule.

Moreover, in the early stages of using the program, certain rule applications that Jape interpreted as *incomplete backward steps* were in fact intended by students as *complete forward steps*, but resulted in misinterpretation by Jape because they had failed to select the appropriate line.

For example, consider conjecture C2:

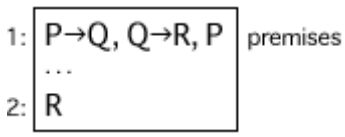


Figure 8: The conjecture C2

A correct procedure would be to click “P→Q” on line 1.1, click the →E rule and so obtain the following display:

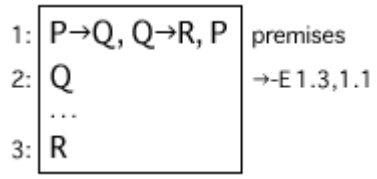


Figure 9: The conjecture C2 after →E on line 1.1

Sometimes, however, students forgot to click “P→Q” before clicking →E; or they clicked “R” on line 2 instead of “P→Q” before clicking →E. In both cases, the following would appear:

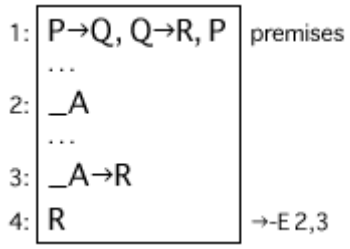


Figure 10: The conjecture C2 after →E without specifying a line

Jape has applied →E *backwards* to the “R” on line 2 and because it has assumed that the user wanted to apply →E but also wanted to defer a decision on the value of the proposition in front of “→R”. In none of the episodes did a student ever indicate that this was his or her intention, and on many occasions, this *backwards* move combined with the *unknown* and the double *ellipsis* caused evident surprise. In such cases, the most common next action was to undo the step, rather than attempting to unify _A with one of the other propositions.

So very often, a missing or inaccurate selection would mislead students into trying the wrong rule. This also had repercussions for ¬I backwards, where the appearance of “_B∧¬_B” would again often be interpreted as indicative of an incorrect step as opposed to an incomplete step.

Another example is when Lewis & Caroline were working on P∧Q ⊢ P (Tape 1, Time 1:00:15). Applying the rule ∧E(L) without clicking “P∧Q” on line 1 results in the line “P∧_B” rather than in the completion of the proof (i.e. the removal of the three dots, and the insertion of the justification “∧-E(L) 1” on line 2).

How do the students interpret this unexpected occurrence? There are at least five possibilities:

1. The program has made a mistake. A consequence of such an interpretation might be asking for help from an expert. This interpretation seems a little implausible, because the students tend to assert greater faith in Jape’s handling of logic than in their own.
2. This is a bad choice of rule. A consequence of this interpretation would be an exploration of other rules. This would indicate a low confidence in their decision-making process with respect to the rules.
3. The rule has been applied wrongly. A consequence of this interpretation would be experimentation with the lines that are clicked before the rule is applied.
4. A slip of the mouse has occurred and an unintended rule has been applied. In this case, one would expect the students to undo and then try the same rule again straightaway, taking care that they are selecting the desired rule.
5. The line “P∧Q” should have been clicked first, because otherwise the possibly erroneous assumption would have to be made that the conjunction that is the subject of the justification for the line “P” is the line “P∧Q”. Jape therefore creates a line containing an unknown, leaving it as an option for the user to point out that the new proposition is, in fact, Q. One might expect talk about the about the justifications or ambiguity.
6. The line “P∧Q” should have been clicked first, for some reason not contemplated too deeply. If Jape asked “Which line do you want to apply the rule to?”, students may realise that a choice is possible, and so think more deeply about this; although such a dialogue would disrupt the elegance of the interface.

It is unclear what exactly Lewis and Caroline concluded here, mainly because of the Left-Right confusion (see below) on the same proof, which complicates the analysis. One of the aims of the Reflection Study is to probe such ambiguous situations.

It seemed to take many students some time to realise that unexpected results were often attributable to a line not being selected before a rule is applied. Perhaps it is because it is not immediately obvious what rule has just been applied. Indeed, it might be felt, by those who did not make use of the facility whereby the rules menu can be "torn off" into its own, permanent window, that the way the menu disappears *instantly* once a rule is chosen is rather intimidating. Maybe highlighting changes to the proof in a different colour, a history of rules, or an indication of which rule is to be undone (as in Microsoft Word, for example) might promote greater reflection on why a particular step did not have the anticipated effect. On the other hand, those students who are of the belief that one will learn all one needs to learn about proving merely by the act of finding a correct sequence of rules will be unlikely to undertake such analysis; and those who, as a matter of course, examine in great detail each step with unanticipated outcomes, will be unlikely to benefit. But such facilities may make reflection easier for those in between these two extremes. It could be argued that the tree display would address this aspect; but there is the obvious danger that by providing too much information (in a perhaps less readable form than the box display) it would confuse rather than enlighten.

This assumption by Jape of a backward step, when students were expecting a forward step, may explain why several students suggested that they were confused about the direction of a rule at certain points in particular conjectures. One such is Rachel (S72). She obtained the best course mark in the cohort. She had a fair amount of programming experience, A-level mathematics, and an interest in learning rigorous methods for programming. Her entire usage of Jape is shown in Figure 11.

Date	Time	Action
Mon Nov 2	15:49:30	Started $P \vdash Q \rightarrow P$
	15:49:51	Completed $P \vdash Q \rightarrow P$
	15:51:08	Started $P \wedge (Q \vee R) \vdash (P \wedge Q) \vee (P \wedge R)$
	15:59:09	Completed $P \wedge (Q \vee R) \vdash (P \wedge Q) \vee (P \wedge R)$
	16:00:04	Started $\neg(P \vee Q) \vdash \neg P \wedge \neg Q$
	16:29:01	Gave up on $\neg(P \vee Q) \vdash \neg P \wedge \neg Q$
Mon Nov 9	15:15:58	New session
	15:17:40	Started $P \wedge (Q \vee R) \vdash (P \wedge Q) \vee (P \wedge R)$
	15:42:41	Completed $P \wedge (Q \vee R) \vdash (P \wedge Q) \vee (P \wedge R)$
	15:43:04	Started $(P \wedge Q) \vee (P \wedge R) \vdash P \wedge (Q \vee R)$
	15:53:59	Completed $(P \wedge Q) \vee (P \wedge R) \vdash P \wedge (Q \vee R)$
	15:54:57	Started $P \vee (Q \wedge R) \vdash (P \vee Q) \wedge (P \vee R)$
	16:11:20	Completed $P \vee (Q \wedge R) \vdash (P \vee Q) \wedge (P \vee R)$
	16:11:35	Started $(P \vee Q) \wedge (P \vee R) \vdash P \vee (Q \wedge R)$
Tue Dec 15	14:11:56	New session
	14:27:13	Started $\exists x.(P(x) \wedge Q(x)), \neg(\exists x.(Q(x) \wedge R(x))) \vdash \exists x.(P(x) \wedge \neg R(x))$
	14:34:40	Gave up on $\exists x.(P(x) \wedge Q(x)), \neg(\exists x.(Q(x) \wedge R(x))) \vdash \exists x.(P(x) \wedge \neg R(x))$
	14:35:35	New session
	14:36:40	Started $\forall x.(P(x) \rightarrow Q(x)), \exists x.P(x) \vdash \exists x.Q(x)$

Figure 11: The Jape usage of Rachel (S72)

Rachel's strong opinion after using Jape for 40 minutes was that she sometimes wasn't sure in what direction a rule would operate. She said that she had particular difficulties with the ∨-E interface, as opposed to the principle of ∨-E, which she understood. She eventually sorted these difficulties out, and it is possible that this involved working out that the premise needed to be clicked before the rule application. Similarly, Caroline makes the comment "It's working opposite to the way I would. It's working backwards." (Tape 1, Time 1:19:40).

One solution to this problem would be to *require* students to select lines, and to have the execution of incomplete steps optionally disabled, at least for →E backwards, ∧E backwards, ∨E backwards and ∨I forwards. No evidence was found of novice students ever wishing to carry out these four steps.

In ¬I backwards, students very often had difficulty in working out or remembering how to unify a variable. A very common error was failing to select the underscore in front of the unknown. An alternative representation of an unknown might be helpful, such an unused proposition (e.g. Z, Y, X) shown in red. A note in a side panel could indicate that Z was an unknown. Another error was to *select* the unknown and the required proposition, rather than to *text-select* them. In this case, the error message "Unify must be given two things to work with" was not always helpful, because the user would be under the impression that he or she *had* selected two things.

Moreover, the actual procedure for unification was found to be tricky: it involves having to hold down one key to text-select the unknown, holding down another key to indicate the selection of a second item, and then text-selecting the required proposition. This multiple text-selection is not a standard part of the Microsoft Windows operating system, and may explain why it appears initially difficult to grasp.

An alternative procedure for unification would be to allow students to double-click the unknown (or right-click or click-and-hold) and then either be presented with a list of choices on a pop-up menu, or be allowed to text-select the required proposition. Such techniques could also provide a general alternative to the passing of parameters. The disadvantage of such an approach is described in the manual: "Formula selection is much quieter than a proffered choice: novices are startled when presented with a choice of alternative actions, perhaps because of the well-known 'focus' illusion: you expect action to occur where you are looking on the screen, even when sometimes that is not the place where the machinery is set to act." (p. 6). However, as has been noted earlier, the problem of users' failure to select the appropriate parameters for rules can introduce unknowns into the proof, and novices appear to be less comfortable about deferring the decision to provide a reference for an unknown than more experienced logicians.

Students tended not to attempt unification prior to Negation unless they were shown the feature by a lab assistant, and so this may have added to the perceived difficulty of the Negation topic. Students repeatedly raised the hyp rule was a feature they did not understand. In particular, it was found puzzling when if "Q→R" and "_A→R" have been selected, they cannot be unified - instead the hyp rule must be used.

One minor graphical design point was raised when it was noticed that one student interpreted the "→-I" rule as "arrow minus one". Although this is perhaps unlikely to be a common error, it is possible that erroneous selections of ∧I instead of →I, for example, may have been due to misreading "∧-I" as "→I". Several students suggested the display of iconic or textual mnemonics for rule structures, which they had difficulty in remembering. It would also be interesting to be able to compare the consequences of using the tree representation that natural deduction proofs can have, instead of the boxes. Certainly greater use of colour and sound to provide character to the different rules and their actions might assist retention of their structure.

A major and common criticism of the interface was that the text in the dialogue boxes that appear when an illegal step is attempted could be made more helpful. The manual points out that "At first the error messages may seem very complicated and rather confusing – indeed, they could do with some simplification – and often the best thing is to read them as if they said 'the rule doesn't apply, so you will have to try something else'." (p. 34). This was exactly how most students treated them. Lewis when faced with a barrage of symbols that constituted Jape's way of telling him that the step he wanted to do wasn't possible - made the poignant aside "It doesn't look like English to me.". There were occasions (such as with the incomplete steps described above) where a more friendly message may have prompted him to reflect a little longer on the reasons why the step failed. There might be a number of ways of reducing the irritation of the error message. There could be a dialogue area rather than a dialogue box; the message could pop-up out of the way of the proof and disappear when something was clicked; the program could use sound to indicate invalid attempts, and provide a help button for further information if desired.

One commonly observed phenomenon is the rapid yet inefficient traversal of the rules menu by the mouse pointer. The most obvious interpretation is that it is akin to doodling, muttering or finger tapping. It is an attempt to while away the seconds while the brain or computer catches up with the task in hand. If so, it is scarcely worthy of attention. Another interpretation is that the student knows what rule they want, but is just having difficulty finding it. A third interpretation is that they are not sure which rule they want and are using the mouse pointer as a visual aid to help them mentally "cross out" potential choices. This is one of the issues that needs to be explored further in the Reflection Study.

3.5 The value of fast, guaranteed accuracy

The guarantee of a correct answer meant that attention could be given to understanding how the rules might be useful. This was of value in helping students to test out their conceptions of the effects and usefulness of rules. It did mean, of course, that Jape prevented students from carrying out illegal steps. It is not clear to what extent should students be prevented by the program from attempting to carry out unhelpful or illegal steps.

On pencil-and-paper, incorrectly positioned boxes often led students astray, even when their basic plan for constructing the proof was initially sound. Using Jape, boxes would always be correct and so this problem never arose. However, knowing which lines the box should encompass is of course an important skill that Jape does nothing to develop. It might be helpful to have a way of allowing students to turn off the box-drawing facility once the basic uses of the rules have been understood, so that boxes have been sketched using the mouse and so box-drawing ability could be developed.

In a similar way, being about to write down the correct justifications for steps is an important skill for pencil-and-paper, and one that is bypassed by Jape, undoubtedly for good reasons in the early stages of learning how to prove. Again, it might be helpful to have a way of allowing students to turn off the justification-writing facility once the basic uses of the rules have been understood, so that justifications have to entered using the keyboard and so justification-writing ability could be developed.

The main advantages of ItL Jape for many students were that it allowed them to consider many more examples than would be possible using pencil-and-paper because of the speed with which proofs could be constructed using the mouse. One disadvantage of this speed was that some steps happened too quickly to grasp what had occurred. However, the undo and redo facilities meant that the step could be replayed at will.

When stuck, students tended to seek help from each other, and from the lab assistants. Kusi and several other students studied the lecture notes. No-one was observed consulting the Jape manual, (either printed or online), although two students were directed to it when they asked about documentation.

3.6 Some rules are harder than others

It was clear from the videotapes that the →I and →E rules were easily handled, in that proofs depending on these are constructed with little difficulty.

A difficulty with the ∧ rules was doubt about whether the rule "∧-E(L)" *selected* or *removed* the left-hand-side of the formula. For example, Caroline - who worked with Lewis for the first workshop - asks, about ∧E, (Tape 1, Time 1:00:15, conjecture P∧Q ⊢ P) "Which one does it keep and which one does it chuck away?". This question fits with the claim that students can be confused about whether the "∧E(L)" rule eliminates the proposition on the left (and so we are left with Q) or whether it eliminates the connective (by *keeping* the proposition on the left, P). Lewis' strategy is "Let's try both of them and see." and this appears to sum up his greater willingness to "undo". Caroline concludes that "It keeps the left-hand-side".

In later work seen on the video, she appears to remember better than Lewis how the (L) and (R) rule suffixes are interpreted by the program; but there is an incident 7 minutes later in which confusion is again apparent. In trying to prove P ⊢ P∨Q, 10 minutes is wasted because they experiment with almost every rule on the menu (except the right one) being applied to almost every line imaginable, because Lewis tried ∨I(R) and not ∨I(L) at first. Caroline does not point out the error, even when Lewis selects the same rule on two further occasions. Perhaps she did not see that he selected ∨I(R), but in that case if she was confident about the correct interpretation of the (L) and (R) why would she not work out from the feedback on the screen that the wrong one had been selected? But even more pressing: given that they had both earlier recognised that left-right confusion as a possible pitfall, why did they not think to check ∨I(L)?

An on-screen history (or other record of what has been tried) might minimise such fruitless experimentation, but, then again, Caroline and Lewis could easily have found some paper to record what they tried but didn't appear to see the need. However, it could be that not being able to recall (without recourse to external means) what one has done over a short time-scale is seen by the students as an embarrassing thing to demonstrate in front of a video camera (they had been videoed for less than 15 minutes at this point), and - more pertinently - more embarrassing than being slow to find solve the logic problem.

Caroline (S107) had maths A-level, a little programming experience, a variety of reasons for studying computer science (although the main one was the prospect of a stable job) scored low on the maths test, well above average on Logic1, the second highest mark on Logic2, and well above average on the exam. Her opinion of the course (just before Logic2) was that it had been very worthwhile, fairly difficult, and interesting. She considered Jape fairly helpful. She spent about 4½ hours using Jape (which counts as "high usage"), across all topics. She proved around 70% of the given conjectures (which counts as "high progress") and 60% of Quantifiers (which was the fourth highest proportion). She did not take part in the Reflection Study.

∧I was found straightforward by most students when the formula containing the ∧ symbol already existed somewhere in the proof, but it was sometimes not clear to the students how to create a particular formula - they were seeking a way to select the two components and then "and" them together (which is not at this time possible using Jape).

Suspicion of the ∨E rule was widespread, to such a degree that students would often prefer to attempt ∨I backwards rather than attempt ∨E forwards. A typical case of this would occur in a conjecture such as P∨Q ⊢ Q∨P. Kusi abandoned it on one occasion. When ∨E *was* used successfully, most students were observed finishing one of the two boxes before starting the other, rather than shifting between the two boxes. In interviews, both Kusi and Lewis described ∨E as being "harder" than the other rules.

It appeared that many students found it difficult to know how to use the negation rules to make progress towards linking the conclusion with the premises. Nearly all students questioned about this suggested that it was important to have "learned the rules" before Jape could be used effectively.

In contrast, the main problem with the quantifier rules was how to get Jape to produce the desired output. In particular, it was not obvious to students how to carry out ∀E or ∃I using a particular variable (i.e. passing a parameter). However, students were clear that they lacked the necessary *conceptual* knowledge to use ∀E, and this was exacerbated by worries about what the appearance of "inscope" meant when it appeared with an ellipsis either side, following the application of this rule or ∃I:

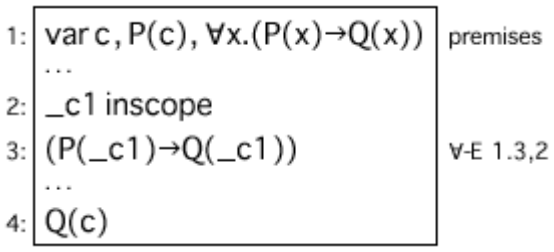


Figure 12: Application of the ∀E rule

With a mid-term test approaching, nearly all students questioned reported that they would be "learning the rules" from the lecture notes in order to prepare for the test, rather than using Jape - they often referred to ∨E, ¬I, ∀E and ∃I. The implication was that it is difficult to use Jape without at least some sort of "grasp" of the rules. This may be related to their ability to interpret the typical output of these rules (see later), and when pressed students suggested that it was not easy to work out what such "difficult" rules did from the output, or how or when they might be useful.

Whether they are correct in this assertion that it is a *conceptual* gap in their knowledge rather than *interface* gap is moot. It could be that students can make little or no progress with the quantifier rules if they do not have a clear idea of how to get the interface to do produce the syntax they expect to see.

In conclusion, then, it would appear that some rules are harder to handle than others. However, this may very well depend on this particular implementation of the natural deduction system

3.7 Judgements of the utility of steps

"Oh no, I'm doing it wrong. (pause) [The proof's] just getting bigger and bigger. (laughs) I'm definitely doing it wrong.".

How soon do students notice things have gone wrong? Several criteria for judging the utility of steps were noted, including looking at the size and number of scope boxes, the appearance of unknowns or variables or "inscope", and the closeness of lines to what is desired. The sudden bifurcation of a proof (two ellipses appear) is another apparently common but unreliable indicator of error. There is little evidence that much attention is paid to the justifications in deciding what rules to apply and where.

Students who are not confident about the effects of particular rules are less likely to be able to distinguish a productive step from an unproductive step. Lewis, for example, appeared to have a strategy at one point of always working backwards and looking for the rules that do not produce unknowns, variables or large boxes.

Mentally checking using informal meanings for the logical connectors that it would be possible to prove a later line from earlier lines was not a common strategy, in spite of much instruction in the semantics of formal logic. Links between the proof rules of natural deduction and the truth-tables of propositions were just not apparent in the discussions of students working at the computer.

3.8 Forward reasoners

There is clear evidence of a distinction between behaviour that is fixated on reasoning forwards and behaviour that is flexible about reasoning forwards or backwards. For example, in the conjecture $P \rightarrow (Q \rightarrow R) \vdash Q \rightarrow (P \rightarrow R)$, a forward reasoner would typically want to "assume P". Perhaps if they had learned from previous experience that this was a dead-end, they might "assume Q". A flexible reasoner might suggest applying →I *backwards*, because this rule generates the assumption.

In Jape, assumptions can *only* be made by applying rules that generate assumptions, such as →I backwards, ∨E forwards, ¬I backwards and ∃E forwards. So forward reasoners will often be uncertain about how to obtain the proof display they know is legitimate.

On paper, students would very often "make an assumption"; whereas in Jape this was replaced by the application of →I backwards. Making an assumption instantly raises the questions "What assumption?" and "What is the conclusion?". Applying →I instantly simplifies the situation, since the assumption and conclusion are automatically generated by Jape:

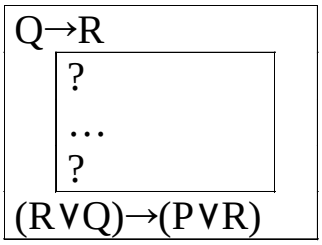


Figure 13: The effect of "making an assumption"

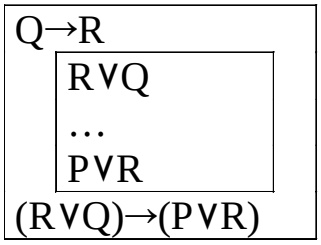


Figure 14: The effect of →I backwards

Conjectures without premises - such as C8 $P \rightarrow (Q \rightarrow P)$ - seem to be viewed as potentially harder a priori. Perhaps it is because there are no premises from which to reason forwards.

3.9 What has been learned can be hard to recall later

During the second workshop, many students started with the conjectures that they had been attempting at the end of the first workshop. These were conjectures with which the students had been starting to have difficulties, and so it was no surprise that many students reported difficulties almost immediately. They were encouraged to go back to conjectures that they had proved easily last time. However, for both Kusi and Lewis, even these proofs were found to be difficult. Kusi eventually recalled enough of the strategies she had been using last time to move on, but

progress in new conjectures was slow. Lewis (who had in retrospect been benefiting from Caroline's heavy guidance) was completely unable to prove conjectures that he had apparently been quite happily proving the week before, and appeared to make no progress. It was not until he worked on the easier proofs in which he had first worked out how particular rules might be applied to certain situations that he was able to make progress.

There are interface issues that might go some way to explain this. For example, it appears to be a not insignificant subtask to work out a way to remember which line of the proof is to be clicked in order to apply a particular rule. However, in explaining the difficulty of recall, it is also possible that the knowledge that students are constructing in order to meet the short-term demands of a particular proof just do not last, for some reason. The pragmatic question then arises of the optimal set of conjectures for revision; the deeper question is why this knowledge is not more long-term.

3.10 A proposed explanatory framework for the findings of the Observational Study

The Popperian psychological perspective developed in Aczel (1998) suggests that learning can be analysed in terms of the trial-and-improvement of strategic theories in response to concerns. Here, the main concern would be to prove conjectures, the theories would be knowledge about natural deduction, and the strategic aspects of these theories would be the heuristics that enable students to prove conjectures. In short, the learning of natural deduction using ItL Jape would consist of the trial-and-improvement of proof strategies.

Although using proof strategies as the main analytic tool is relatively uncontroversial, other approaches would assert that there are other, perhaps more sophisticated explanatory frameworks that could be employed in addition to the analysis of proof strategies. For example, a neo-Piagetian perspective might consider the maturing of decontextualised cognitive structures; a Vygotskian perspective might consider language and the software in their role as mediators of learning; a cognitive science perspective might analyse the task demands of the different rules; a behaviourist perspective might consider the optimal sequence of proof presentation that best produces improvements in observable proof success rates. Such analyses could be carried out in future, but an analysis in terms of proof strategies is offered as a start.

3.10.1 Strategic knowledge

Some of the more or less readily identifiable strategies that students might be using to help them construct proofs are *rule-specific*, such as «If there is an arrow as the principal operator in the conclusion, break up the conclusion using →I.».

Clearly not all students would, if asked, inevitably come up with the same linguistic formulation for this →I strategy, but it is an attempt to capture in words the flavour of a strategy that appears to account for very many student actions and that is repeatedly articulated by students in similar terms to these. Note that there are likely to be subtle variations. For example, it might be that algebraically-inclined students think of applying →E to P→Q as "substituting" P into the function-machine formula (a common metaphor in school mathematics) to get Q. →E is seen as a functional operator rather than as a relational rule. Or it could be that students are operating purely syntactically (something like «Given that P is a line of the proof and that P→Q is a line of the proof, the rule →E allows the line Q to be written.»). Or it could be that students are using an informal notion of existential proof (something like «If a proof of P exists, and a proof of P→Q exists, then that is sufficient to prove Q, justified by the axiom →E.»). Or it could be that they are using a notion of truth (something like «P→Q tells me that if P is true, then Q is true. But P *is* true, so Q is true. →E is the instruction to point this out.»).

Typically, it seems that there are few indications from students' talk in this study of which variants might be being used. Therefore, it is hoped the Reflection Study will yield more clues in this regard. By tackling selected proofs, and being asked to describe their interpretations of what they are doing, the aim is to explore more closely students' strategies for constructing proofs. However, one must then be aware of the likelihood of obtaining post hoc rationalisations rather than definitive cognitive mechanisms.

Some of these rule-specific strategies help students to choose between rules; for example, «If there is a choice between ∨E forwards and ∨I backwards, try ∨E first.». The student Kusi described this as the "precedence" of ∨E

over ∨I. Other rule-specific strategies seem attempts to evaluate the success of a particular course of action before it is attempted; for example, «Proof by contradiction ¬E¬I may be useful in the case ¬A⊢B (where A and B represent complicated formulas) if A would be easier to break up than B.».

In contrast to rule-specific strategies, there appear to be strategies that might be called *global strategies*. Three important examples would be «When reasoning forwards, check if the lines produced are useful in obtaining the conclusion.»; «When reasoning backwards, check if the lines produced are provable from the premises.»; and «The principle operator in a line is the only operator that determines the legal rules applicable to that line.». A conjectured strategy for which no evidence has yet been seen is «If stuck on a conjecture with no premises, try to think of a previously-proved theorem that could be applied, so as to allow forward reasoning.».

An important global strategy that is only applicable under Jape would be «For each ellipsis, the line directly below the ellipsis is the only conclusion that needs to be proved.».

3.10.2 Conjunction strategies

For the conjecture (P∧Q)→R ⊢ P→(Q→R), some students, after →I backwards twice, look for a way to conjoin two propositions. For example, Caroline is heard to ask , "How do you get the 'ands' to come into it?" (Tape 1, Time 1:02:29). One can imagine therefore that the strategy of something like "To create 'P∧Q', find P (on line *x*, say), find Q (on line *y*, say) and write down ∧I, *x*, *y*" is forced, by using Jape, to change to something like "To create 'P∧Q', find a more complex line containing it, and find some rules that break the line down". This change is a direct result of the philosophy "You don't have to introduce arbitrary assumptions ever: all that is necessary is that sometimes you are prepared to work backwards – from conclusion towards assumption – rather than always forwards. The gain is that when you introduce an assumption in this way you never have to guess what it might be." (from the "Using ItL Jape" manual, p. 4).

So, in fact, although students might perceive their difficulty with this conjecture to be to do with ∧I (because it is not clear if Jape will let them "and" the P and Q together), it could be interpreted as to do with → (because Jape expects 'P∧Q' to be made this way). The strategy for →E mentioned earlier ("Whenever you see an arrow as the principal connective in a line *before* a gap in the proof, check if the LHS is a line in the proof; and if so split the line using →E") should be amended by students so that "check if the LHS is a line in the proof" becomes something like "check if the LHS can be proved from the premises".

It is clearly the aim of this implementation of Jape to encourage such favoured ways of working, as opposed to allowing students to wallow *around* in a mire of irrelevant assumptions. Even so, some students were heard asking lab assistants "How do I make an assumption", although I don't think there are any examples of that on any of the videos.

3.10.3 Disjunction strategies

As discussed earlier, ∨I was used successfully on P ⊢ P∨Q only once the left-right confusion was sorted out. It is possible that the strategy developed in the end was something like "To prove P∨Q, either P or Q must be on a previous line; so check that at least one of them is there and use ∨I. If P, use (L); if Q, use (R); and if both, use either." . However, this strategy has the same disadvantage as given above for ∧I: it might be possible to prove the P or Q.

The conjecture P∨Q ⊢ Q∨P was found hard by many students, because, as has been discussed before, their inclination seems to be to use ∨I on Q∨P, just as they did for P ⊢ P∨Q and Q ⊢ P∨Q - and perhaps by analogy with →I. Like many others, Lewis and Caroline (Tape 1, Time 1:18:30) seemed not to check instinctively whether a line containing P is useful. When they do eventually try ∨E, they do not immediately undo once they see the double scope boxes. This, I think, contrasts nicely with Kusi's actions in her second use of Jape (Tape 1, Time 1:30:00) when tackling Q ⊢ P∨Q and she initially selects ∨E: whether it is the variables _A and _B, or the double scope boxes, or the sudden expansion of the proof, she immediately undoes the rule. A few minutes later, she says "I can't do these 'ors'.", in the context of ∨E (rather than ∨I).

The issue of how Jape chooses the final line in the scope boxes for ∨E did not appear to arise in the episodes viewed so far. In fact, we did not seem to get much insight at all into Caroline's and Lewis' understanding of the ∨E rule. But it seems clear that Lewis is not *purely* following instructions from Caroline. There are several

occasions in these proofs where he can choose the rule successfully, so long as Caroline tells him which line to click (and, if necessary, whether it is the right or left rule).

3.10.4 The development of the strategies

It could be argued that these strategies represent no more than an *ad hoc* collection of "rules of thumb" that demonstrate little of the deep understanding that an experienced logician might have, and show little regard for the circumstances in which they might fail. Moreover, it might be asserted, using a proof assistant program actually *encourages* a "blind", purely syntactical, pattern-matching trial-and-error approach - for example, «Look for the 'main symbol' in the most complicated line. Find the same symbol in the list of rules. Try one of the matching rules. Undo the rule if the display doesn't look right (criteria for which might include any large, unexpected increases in the length of the proof, the number of boxes, the number of gaps in the proof, the appearance of unfamiliar symbols); and try another.». When this strategy fails, all that students have left is the degenerate strategy «Click on one of lines. Keep on trying rules. Try a different line. ».

This argument is supported by video evidence that the student Lewis had some difficulty in recalling proof strategies that were apparently strongly founded a week before. The strategies that students are constructing in order to meet the short-term demands of a particular proof just do not appear to last, for some reason; and it is possible that the reason is superficial understanding. Could learning through trial-and-error of rules be a faster route to *ad hoc* tactics? It is also possible that even where understanding is thorough some sort of instructional intervention is desirable in order to support later recall.

On the other hand, while it has to be admitted that many of the simpler conjectures succumb to such a primitive trial-and-error strategy, it would have to be asked, then, where the more sophisticated strategies described above come from. There is little evidence in this study that they come from instruction; where students are using pattern-matching trial-and-error at the start of a set of exercises they often end up articulating or at least demonstrating the more sophisticated strategies. We would argue that students are *developing these strategies for themselves*, and, furthermore, they are developing them *precisely because pattern-matching trial-and-error is progressively found to be inadequate*.

Moreover, portraying these strategies as purely *mechanical* responses to a limited set of straightforward syntactical inputs seems undeserved. Not only do the strategies to be applied at a particular point in a proof have to be selected with care - particularly in conjectures involving negation or quantification - and not only do the strategies have to be adapted progressively as counter-examples are encountered, but the strategies also clearly incorporate expectations about what a proof should look like, about why a particular rule might be applicable in certain circumstances, about what might or might not be provable, and so on.

Finally, the role of the software in this is not so simple as encouraging pattern-matching trial-and-error. We did find evidence of the latter, and it was even noted by one student "He doesn't know what he's doing. … He's proving them but he doesn't know what's going on."; but early indications from the analysis are that success with mechanical methods was short-lived. Whether a more reflective approach is more or less likely when using software or using pencil-and-paper is at this stage uncertain; but what does seem clear is that for many students, using ItL Jape allowed them to consider many more examples than would be possible using pencil-and-paper (because the program takes on the task of drawing the proof) and it also guaranteed that inadequate proof attempts and incorrect rule applications were immediately challenged. It would therefore be reasonable to suggest that for those students adopting a reflective approach, the development of sophisticated proof strategies would be more effective when using the software than when using pencil-and-paper.

Indeed, several students expressed the view that, although it was not easy to work out from the program what the "difficult" rules did, or how or when they might be useful without at least some sort of initial "grasp" of the rules, the main advantage of ItL Jape was that it allowed experimentation in order to work out proof strategies. It would be interesting to be able to identify what precisely this "grasp" might be and why the feedback provided by the program was insufficient to provide it. That some rules might be harder to use than others suggests that it seems to be harder to construct useful strategies for some proof rules than others, given the particular implementation of the natural deduction system. But why might this be?

3.11 Implications for the Reflection Study

Each of the findings (as opposed to the explanations for the observed findings given in section 3.10) in the Observational Study is inspired directly by one or more incidents. The main aim of the Reflection Study was to test these findings by putting students in similar situations to the original incidents, seeing if the incidents are replicated, and, if so, obtaining students' interpretation of the incident. Such interventions were not part of the Observational Study because they would have disturbed the natural flow of the students' work and directed attention to aspects of the situation that might not otherwise be noticed.

In addition, the Observational Study has been extremely useful in enabling the identification of a number of possible strategies that students might be using to help them construct proofs, and another aim of the Reflection Study is therefore to investigate more thoroughly the most common proof strategies, including the priorities for different rules, and behaviours when the priorities fail. Most importantly, it is hoped to find out in more detail how ItL Jape can assist in the improvement of strategies. one key strategy would appear to be semantic checking of provability. Whether semantic checking became more common as the conjectures become harder to prove is not yet clear, and this is something that needs to be explored further in the Reflection Study.

It was noticeable from the videotapes that students' comments about the *nature* of the activity of constructing proofs were lacking. It is not clear whether this is because students were absorbed so completely in the activity that the issue was not considered, or because the issue was so central to the activity itself that linguistic formulation of an opinion was too difficult. In either event, obtaining students perceptions of what they are doing is also an aim of the Reflection Study.

The Observational Study has provided largely qualitative data relating to usage of ItL Jape. From the Measurement Study - which took place in parallel to this work - there is also some quantitative data available from the logfiles which recorded most students' usage of the program, some survey data on background factors that might be relevant to program usage, some test data which might indicate what students gained from using the program, and survey data on students' experiences of using Jape. The Measurement Study is reported next.

4. The Measurement Study

Largely quantitative data was collected relating to logic students' usage and opinions of ItL Jape, relevant background factors, and various course outcome measures. This section reports on the analysis of this data.

4.1 Data Collection

In this study, initial profiling of the students enrolled is intended to provide a baseline for evaluating Jape's effectiveness using performance indicators taken throughout the course. The use of control groups was not possible, on both ethical and pragmatic grounds, so an analysis of the comparative effects for different backgrounds of the student population must be carried out.

The data collected includes results and sample scripts from a variety of sources:

1. *Survey1*: a survey given to students early in the course, eliciting background information on the students (such as A-levels and prior computer experience), motivations for choosing to study computer science, and expectations of the logic course.
2. *Maths Test*: a voluntary test of school-level mathematics taken by some of the students prior to the start of logic course.
3. *Jape Usage*: a history of students' interactions with Jape, as recorded in logfiles.
4. *Survey2*: a survey asking for students' views about their experiences of the course and of Jape.
5. *Logic1*: a written outcome measure, set as part of the course, primarily on propositional logic. About a third of the test involves constructing natural deduction proofs.
6. *Logic2*: a written outcome measure, set as part of the course, primarily on predicate logic. About a third of the test involves constructing natural deduction proofs.
7. *The exam*: a written outcome measure, set as 80% of the assessment for the course, containing a mix of propositional and predicate logic. Up to a third of the test involves constructing natural deduction proofs, however there was a choice of questions that meant that students could avoid natural deduction proofs completely.

4.2 Survey1 - Background Data

This section gives a brief overview of the analysis of the first two sections of Survey1. This firstly asked for basic background information from students (such as A-levels and prior computer experience), and secondly tried to find out students' motivations for choosing Computer Science and their expectations of the logic course. This information will be used for profiling the students with respect to outcome measures. The third section of Survey1 - a selection of reasoning items - has not been included in this analysis.

Some data for students who did not complete the survey, and some variables not included as items in Survey1 have been supplemented with information obtained through the university. For the convenience of reporting the results, all this data has been included in this section.

The survey form is in the appendix to this document.

4.2.1 Survey1 - sample

The survey was taken by 149 students during the second lab workshop of term. This is roughly 80% of all students associated with the course at some point during the term.

24% of the students in the sample are female. All but 7% (11 students) are under 25 years old. Of these older students, 5 are male and 6 female. 64% are registered for the degree "Computer Science"; 19% are registered for "Computer Science & Mathematics" or "Mathematics & Computer Science"; 13% are registered for "Computer Science & Business". About a quarter of the sample are overseas students.

4.2.2 A-Levels

According to the students' responses, 81% of them have mathematics A-level. Of the others, there are several students who have undertaken some mathematics at that level, in courses such as A/S mathematics, the BTEC Diploma in Computer Studies, the French Baccalaureate and other overseas qualifications; the majority of these students, however, have not.

The most popular A-level is mathematics, followed by chemistry (43%), physics (33%) and biology (27%). Just under a third of students say they have a computer-related A-level or equivalent (such as "Computer science", "Computing", "IT", "Computer Studies", or "BTEC National Diploma in Computer Science"). Economics and business studies are the next most popular.

The most popular A-level combination is mathematics-biology-chemistry, which was taken by a fifth of students; the next most popular combination was mathematics-physics-chemistry (13%).

4.2.3 Prior Computer Experience

Over three-quarters have had little or no experience of programming before they start the course. 42% indicated that they have *no* prior programming experience; 34% indicated that they have "a little" programming experience; 15% indicated that they have "a fair amount" of programming experience; and 9% indicated that they have "a lot" of programming experience.

A slightly higher proportion of females than males had little or no programming experience.

92% indicated that they have had use of a computer at home before they came to university. Of these, 70% played games on it; 52% used spreadsheets; and 31% used it for programming.

4.2.4 Motivations

Students were asked to list *all* the important factors in their choosing to study a degree that includes computer science. Four factors were chosen repeatedly: "learning to program" (73%), "following an interest" (73%), "financial rewards" (75%) and "stable job" (71%).

40% indicated that "learning to think logically" was a relevant factor in their decision. A higher proportion of females (53%) than males (36%) chose this factor.

Very few selected as relevant factors "learning rigorous methods" (18%), "friends also studying it" (7%) or "other" (3%). Only 5% (7 students) admitted that "pressure from parents" was a factor, and only 13% (20 students) that they "couldn't think of anything else to study". These students nearly all lacked a computer-related A-level and had little or no prior programming experience.

The student groups that were more likely than other groups to chose "financial rewards" as a relevant factor were: males, students registered for the degree "Computer Science", and non-overseas students.

The student groups that were more likely than other groups to chose "stable job" as a relevant factor were: students with no programming experience, and students who haven't used a computer at home. There may be slight effect for students without a computer-related A-level and students registered for the degree "Computer Science & Business Studies".

The student groups that were more likely than other groups to chose “learning to program” as a relevant factor were males, older students (10 out of 11), students without mathematics A-level, students with a computer-related A-level, students without A-levels (24 out of 25), students registered for the Computer Science degree (students registered for “Maths & Computing” were least likely to chose this factor), and students who have used a computer at home.

The student groups that were more likely to chose “following an interest” as a relevant factor were older students, students without mathematics A-level, students with a computer-related A-level, students without A-levels, students with some programming experience, and students who have used a computer at home. The more prior programming experience, the higher the proportion of students selecting “following an interest” as a relevant factor.

The factors could be roughly divided into those which are *intrinsic* in the sense of being primarily about the value of engaging in the subject matter (such as “learning to program”, “following an interest”, “learning to think logically”, or “learning rigorous methods”), and those that are *extrinsic*, in the sense of being primarily about the value of the course as a means to an end lying outside the subject matter (such as “financial rewards”, “stable job”, “pressure from parents”, or “friends also studying it”).

84% of students had *both* an intrinsic reason *and* an extrinsic reason for studying Computer Science. Males were slightly more likely to have an extrinsic reason.

Students were also asked to indicate which was the *most* important factor in their decision. 40% indicated “following an interest”; and 30% indicated “financial rewards”. Much fewer emphasised “learning to program” (13%) or “stable job” (13%, virtually all being without a computer-related A-level). Excluding the 15 students who did not indicate a most important factor, these four factors accounted for all but 7 students.

The student groups that were more likely than other groups to chose “financial rewards” as the most important factor were: students with little or no prior programming experience ”(none of those indicating “a lot” of prior programming experience selected “financial rewards” as the most important factor - most were “following an interest”), and students registered for the degree “Computer Science and Business Studies”.

The student groups that were more likely than other groups to chose “stable job” as the most important factor were: students without a computer-related A-level, and students without a home computer.

The student groups that were more likely than other groups to chose “learning to program” as the most important factor were: students without A-levels, and overseas students. None of the students registered for the degree “Maths and Computing” chose this factor.

The student groups that were more likely than other groups to chose “following an interest” as the most important factor were: students with a computer-related A-level, and students with some programming experience (the more prior programming experience, the higher the proportion of students selecting this factor).

The student groups that were more likely than other groups to have an *intrinsic* main motivation for studying the degree were: students with a computer-related A-level, and students who have a home computer. A strong trend is that the more prior programming experience students have, the more likely they are to have an intrinsic reason:

Programming experience	N	% with an intrinsic main motive
none at all	56	45%
a little	44	52%
a fair amount	20	80%
a lot	13	92%

Figure 15: Proportions of students with intrinsic motivations, by programming experience

Students registered for the degree “Computer Science and Business Studies” were slightly more likely than other groups to have an *extrinsic* reason for studying the degree.

4.2.5 Course Expectations

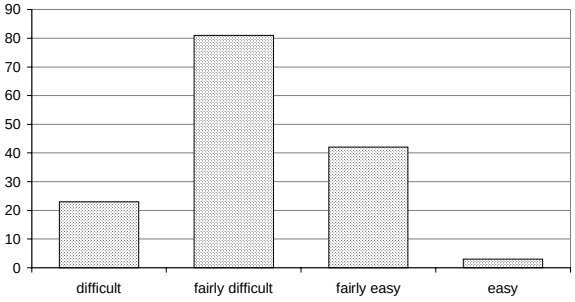


Figure 16: Responses to "How difficult are you expecting the logic course to be?"

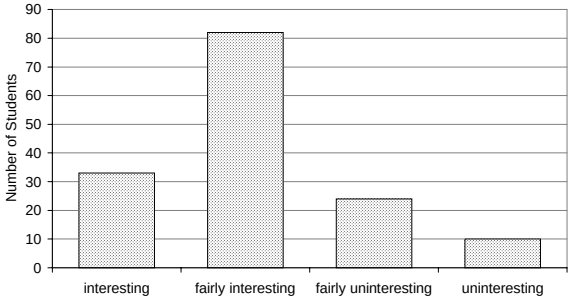


Figure 17: Responses to "How interesting are you expecting the logic course to be?"

A higher proportion of the females than the males expected the course to be "difficult" or "fairly difficult". All but one of the students registered for the degree "Maths & Computing" expected the course to be "difficult" or "fairly difficult".

None of the older students, BTEC students, Access students, students registered for "Computer Science & Maths" or students with a lot of programming experience expected the course to be "uninteresting" or "fairly uninteresting". All but one of the students with a computer-related A-level expected the course to be "interesting" or "fairly interesting".

Responses to the question "What is the value of the 'Introduction to Logic' course, as you see it?" could be roughly divided into 4 categories, depending on the terms in which the value is perceived:

(i) a training in rational, valid, efficient, unambiguous or methodical thinking (e.g. "[The course] helps you to think logically"; "To argue more precisely"; "Develops your mind"; "An alternative way to approach problems"; "a new way of thinking");

(ii) an introduction to the theoretical "foundations" of Computer Science, of programming or of mathematics (e.g. "Everything in Computer Science is based fundamentally on logic"; "To ease you into the fundamental and basic concepts of Comp. Sci.");

(iii) a preparation for programming (e.g. "To be taught how to sum up the operation of a program in a way that others may understand."; "This course helps you to think in the same way that computers think"; "To help us to learn effective computer programming skills"; "To help with program development"; "Being able to see why errors in programs… occur due to logical steps not being followed through properly");

(iv) of little value (e.g. "I don't find it very useful"; "So far I don't see any connection in the course with computer science"; "no idea so far")

Around a third of students left this question blank.

4.3 Maths Test

This section gives a brief overview of the analysis of the maths test taken by students prior to the start of the logic course.

4.3.1 Maths Test - sample

125 students associated with the logic course took the maths test, roughly 70% of all students associated with the course at some point during the term.

The test was voluntary, but tutors strongly advised those of their students who might have difficulties with mathematics to take it. Judging by the proportions of the student groups identified in Survey1 - for students who took both the maths test and Survey1 - the sample for the maths test is similar to that for Survey1.

4.3.2 Maths Test - % score

The mean score was 60%, with standard deviation 20.5.

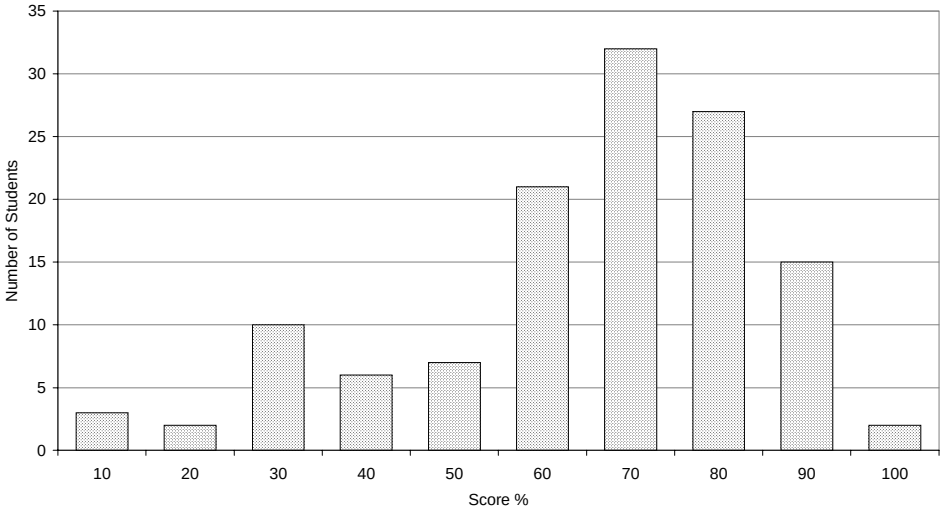


Figure 18: Student Maths scores

4.3.3 Maths Test - scores by student group

The mean scores for the different student groups identified in Survey1 are fairly similar to each other, with 5 exceptions:

1. The mean score for those with mathematics A-level is higher than for those without, as one would hope (67% for 95 students compared to 38% for 24 students; $p<0.001$ for a 1-tailed t-test with unequal variances assumed). This factor can account for at least a third of the total variation in scores.
2. The mean score for those with a computing-related A-level is *lower* than for those without (50% for 38 students compared to 66% for 81 students; $p<0.001$ for a 2-tailed t-test with unequal variances assumed). Prior programming experience does not seem to have an effect.
3. The mean score for those who indicated "financial rewards" as a relevant factor in their decision to study a degree involving computer science is lower than for those who didn't (59% for 86 students compared to 71% for 24 students; $p=0.006$ for a 1-tailed t-test with equal variances assumed). However, the difference for those indicating the "financial rewards" as the *most important* factor was not significant.
4. The mean score for overseas students is *higher* than for home students (69% for 27 students compared to 58% for 97 students; $p=0.002$ for a 2-tailed t-test with unequal variances assumed).
5. The mean score for those who expected the logic course to be "difficult" or "fairly difficult" is lower than for those who expected it to be "easy" or "fairly easy" (57% for 74 students compared to 72% for 36 students; $p<0.001$ for a 1-tailed t-test with unequal variances assumed).

4.4 Jape Usage

Students were observed and videoed using Jape as part of their regular workshops on the logic course. Students' usage of Jape was also automatically recorded by the program in logfiles. This section reports an analysis of the logfiles - supplemented with relevant observational data - in order to provide an account of students' experiences with the program.

4.4.1 The logfiles

Students were made aware of the logging mechanism and the purposes to which the data would be put. It was made clear that the logfiles were to be used for research into the program, and would not be used for assessment of students. Clearly, however, telling students that a record would be kept of how often and for how long they use a particular program might influence some students' subsequent behaviour, but this is unavoidable.

A short extract of a logfile (slightly tidied up) is shown in the following figure.

```
Mon Nov 2 15:24:55 GMT 1998 OPENPROOF {P ⊢ P∨Q} 8
Mon Nov 2 15:25:22 GMT 1998 COMPLETE
Mon Nov 2 15:25:28 GMT 1998 CLOSEPROOF 8
Mon Nov 2 15:25:28 GMT 1998 INCOMPLETE
Mon Nov 2 15:25:33 GMT 1998 OPENPROOF {Q ⊢ P∨Q} 8
Mon Nov 2 15:25:44 GMT 1998 COMPLETE
Mon Nov 2 15:25:48 GMT 1998 CLOSEPROOF 8
Mon Nov 2 15:25:48 GMT 1998 INCOMPLETE
Mon Nov 2 15:26:03 GMT 1998 OPENPROOF {P∨Q ⊢ Q∨P} 8
Mon Nov 2 15:31:07 GMT 1998 OPENPROOF {Q→R ⊢ (P∨Q)→(P∨R)} 9
Mon Nov 2 15:46:47 GMT 1998 FOCUSPROOF 7
Mon Nov 2 15:47:09 GMT 1998 CLOSEPROOF 7
Mon Nov 2 15:47:34 GMT 1998 FOCUSPROOF 8
Mon Dec 7 14:17:13 GMT 1998 SESSION it058
Mon Dec 7 14:17:15 GMT 1998 INCOMPLETE
Mon Dec 7 14:20:34 GMT 1998 OPENPROOF {var c, P(c), ∀x.(P(x)→Q(x)) ⊢ Q(c)} 1
Mon Dec 7 14:21:28 GMT 1998 OPENPROOF {∀x.(P(x)→Q(x)) ⊢ ∀x.P(x)→∀x.Q(x)} 2
Mon Dec 7 14:44:51 GMT 1998 COMPLETE
Mon Dec 7 14:47:31 GMT 1998 OPENPROOF {∀x.(P(x)→Q(x)), ∀x.(Q(x)→R(x)) ⊢ ∀x.(P(x)→R(x))} 3
Mon Dec 7 14:47:31 GMT 1998 INCOMPLETE
Mon Dec 7 14:54:48 GMT 1998 COMPLETE
Mon Dec 7 14:57:01 GMT 1998 OPENPROOF {∀x.P(x)∧∀x.Q(x) ⊢ ∀x.(P(x)∧Q(x))} 4
Mon Dec 7 14:57:01 GMT 1998 INCOMPLETE
```

Figure 19: Extract from a logfile - Kusi (student S18)

It should be noted that some students may be using Jape at home, for which timings would not be available. It has also not been possible for this research to obtain more detailed records of the particular rules applied by students during proving - ideally, a complete record of each student's interactions with the program would be captured.

There is an important distinction to be made at the outset between a *conjecture* (of which there are 70 built into ItL Jape) and a *proof attempt* - which constitutes an attempt by a student to prove a conjecture for a particular period of time. All the conjectures are different; but a proof attempt can be on a conjecture that has already proved, or that was abandoned previously.

A student's work can also be divided into *sessions*, which start when the student runs ItL Jape, and finish when the student exits ItL Jape. Sessions may be immediately consecutive (if the system crashed, for example), later in the day (if student decided to work in his or her own time after a lab workshop, for example), or on a different day.

In the statistical analyses that follow, the Mann-Whitney U test has been used where the variable in question does not appear to be from a normal distribution. It is equivalent to the Wilcoxon rank sum test and the Kruskal-Wallis test for two groups. It relies on the two distributions being similar in shape.

4.4.2 Jape Usage Sample

There are on record 178 students associated with the course at some point during the term (although some of these may have early on switched courses or dropped out of the degree). There are 146 logfiles relating to these students (82% of the total). In addition, there are 27 students (15%) for whom there was apparently no logfile and a further 5 students (3%) for whom it was not possible to determine if they had a logfile.

However, of these 146 valid logfiles, 21 showed no Jape usage at all. Once 3 logfiles that contain almost no activity have been excluded (i.e. no completed proofs and no more than one attempted proof), there are 122 students who used Jape. So there is evidence therefore that at least 70% of all students used Jape. It is also possible that some students worked with a friend or on a home computer.

For the purposes of this analysis, those students without a logfile - or for whom it was not possible to determine if they had a logfile - will be excluded from analysis. Due to an error at registration time, logfiles were mostly not recorded for students registered for “Maths & Computer Science”.

Of those who completed Survey1, 90% used Jape. The program was used by a similar proportion of those who took Logic1, Logic2, the exam and the maths test. There is data for all six instruments for 81 students.

Jape was used in three compulsory lab workshops, and could be used by students in their own time. So those who did not use Jape may, for example, have dropped out of the course, or they may have been ill, or they may simply have chosen not to attend for whatever reason. However, it was observed that there were fewer students in the second lab session than the first, because students could choose instead to attend revision classes being run at the same time. There were more students in the third than the second, but not as many as the first. In the third lab session, it was clear that at least a third of them had been working on Jape outside lab sessions.

4.4.3 Analysis of the logfiles - technical notes

When a proof is completed or abandoned, the following data can be extracted from the logfiles: the student’s userid, the conjecture being attempted, the time spent on the conjecture, the date, and whether the conjecture was completed. The steps the student takes within a proof are not available.

Multiple attempts at the same conjecture are consequently recorded as separate proof attempts. However, if, while working on a particular proof, the student moves onto another proof without dismissing the window of the first proof, and then *returns* to that window later in the same session, the interactions within the same window are treated as all one proof attempt.

A student can infrequently attempt the same conjecture in two different windows. Once it has been proved in one window, the student will typically abandon the proof in the other window, perhaps because they consider it of little importance to proof the same conjecture twice. One consequence is that he number of *proof attempts* that are unsuccessful is overestimated (the number of *conjectures*, of course, is accurate).

Under the Unix version of the program that students were using, when a proof window is closed, no proof is visible (under the MacOS version of the program previous proofs are visible).

One main disadvantage of the logfile data is that for proof attempts that are recorded as taking some time, it is not known if the student was actually working on the proof the whole time (because of toilet breaks, chats with friends, dealing with emails, for example). Proof attempts with times over 30 minutes are flagged by the analysis software - there are 10 such proof attempts. The longest is 79 minutes, for student 22 while working on the conjecture $P \vee (Q \vee R) \vdash (P \vee Q) \vee R$. However, it is not obvious that a student would *not* spend over an hour on this proof. The second longest proof is 68 minutes, and it is for the same student and the same conjecture, one day later. (This student, aged over 25, obtained a grade D for the course as a whole. These two proofs accounted for almost half of his total time spent using Jape. The logs show that he did not get much beyond $\vee$E.) The next two longest proofs (62 minutes and 58 minutes) are for student 7. Again, there is no obvious way to tell whether the student was actively working on these proofs, or even sitting at the computer.

Another disadvantage of the recording mechanism is that the time of the end of each session is not recorded in the logfile. As a consequence, if an incomplete proof is visible at the end of a session, it is not known when the student finished working on that proof - the time recorded as spent on such proofs is underestimated.

The logfile mechanism records when ItL Jape has detected that a given proof attempt is complete. It does not record when the student marks the proof as complete (and often students have been observed not bothering to do so, in any case, even when they are aware that the proof is complete). Sometimes proofs marked complete become

incomplete. This is likely to be due to the student "undoing" the final step, perhaps to explore the reasons why the step finished the proof. Undoing is not explicitly recorded in the logfile, but future research might want to consider the values of such exploration as a learning strategy. For the purposes of the current research, the proof attempt is regarded as complete when the proof is first identified as complete by ItL Jape, and further effort by the student on the same proof window are ignored. This is reasonable for the vast majority of proof attempts, especially since only a few students explored in this way.

The total time and number of conjectures recorded by three students who participated in the research (S18, S107, S108) have been incremented to allow for when they worked using the research userid - this userid allowed better quality video recording (the screen refresh rate could be set) but it failed to record Jape usage in logfiles.

The recalling of a file of previous proofs is not recorded in the logfile, and so the figure for the number of proofs that are attempted is an overestimate. An approximate method of estimating when this recalling occurs has been developed - see below.

The total number of proofs in all the logfiles relating to students that are part of the study is 4919. Of these, 3260 proofs are marked complete. 1659 are marked incomplete.

Not all of the 3260 proofs marked *complete* were in fact tackled by students. This is because some completed proofs were either selected for review or else recalled in a file of previous proofs. Therefore, the records of completed proofs that apparently took zero time (187 proofs) have been removed. For the same reason, clearly complex proofs that apparently took up to two seconds (26 proofs) have been removed. Future research might want to consider which students used the review facility as a learning tool. This leaves 3047 proofs which students tackled and completed.

Not all of the 1659 proofs marked *incomplete* were in fact tackled by students. This is because some incomplete proofs could have been recalled in a file of previous proofs, and then abandoned at the end of the session without the user seeing them (under Unix, at least). This explains why there are records of incomplete proofs that apparently took zero time, and why these proofs are all recorded as having been abandoned at the end of the session. However, some of these times are zero because they relate to proofs that were recalled but not tackled, and some are because they relate to proofs that *were* tackled but at the end of a session (when the end time is not recorded). So at least some of these 650 proofs should be removed, but it is not clear which ones. There are a further 724 proofs which were abandoned at the end of the session but which have non-zero times recorded.

In order to correct for the problem of the recalling of saved proofs, an approximate method for assessing when this occurs has been implemented. This suggests that of the 1659 proofs that are marked incomplete, 391 were recalled rather than selected. So 1268 proofs were tackled by students but not completed.

This analysis leaves 4315 attempted proofs. Of these, it has been calculated that 364 proofs were abandoned incomplete but visible at the end of a session. So the times for these are underestimates.

4.4.4 Who used Jape?

There are records of 122 students actively using Jape. Of the students who took at least one of the tests and for whom logfile data is available, 86% used Jape at least once (120 out of 139 students).

All of the females used Jape (compared to 80% of males); and all of the older students used Jape (compared to 83% of those under the age of 25).

Other groups that were slightly more likely to use Jape are students with some programming experience, students for whom "following an interest" was a relevant factor in their choosing to study Computer Science, and students with no extrinsic motivations. The more difficult the course was expected to be, the less likely using Jape became. The more dull the course was expected to be, the less likely using Jape became.

Students who used Jape were much less likely to be absent from tests (including maths, Survey1, Logic1, Logic2 and the exam) than students who did not use Jape.

4.4.5 How much was Jape used?

Student statistics	Number of attempted proofs	Number of successful proofs	Minutes spent using Jape
Mean	36	26	86
Median	26	21	59
Std. Deviation	33.1	23.5	84.5
Minimum	2	0	4
Maximum	188	151	443
Interquartile Range	32	25	77
Skewness	2.1	2.4	2.0
Kurtosis	5.4	8.5	4.4

Figure 20: Summary data for Jape usage

Note: these figures are mostly unchanged if the 364 proofs with underestimated times are removed (that is: the proofs that were abandoned incomplete but visible at the end of a session).

Of the students who used Jape, the median number of proofs attempted was 26, but the range was 2 to 188. Judging by the number of proofs attempted by each student, Jape usage is not distributed normally:

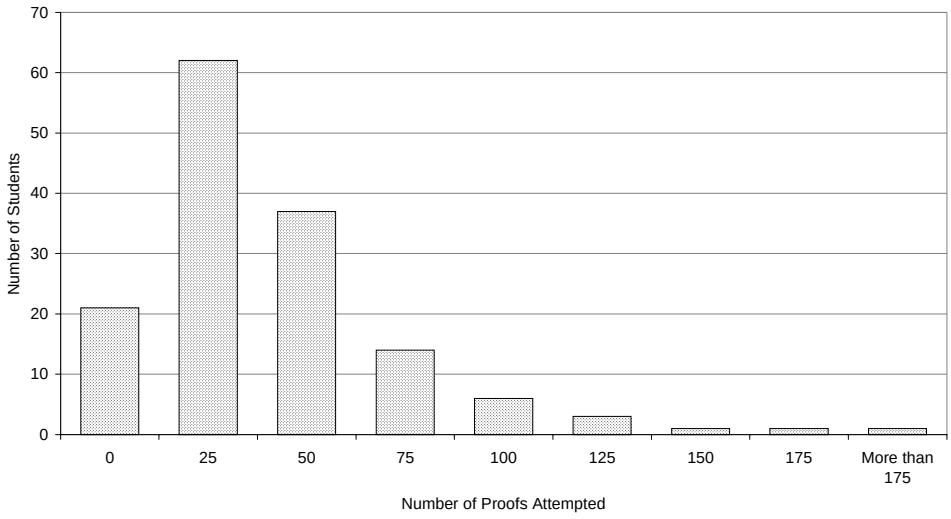


Figure 21: Frequencies of total number of proofs attempted by students

Jape usage as measured by time is similar to that as measured by the number of proofs:

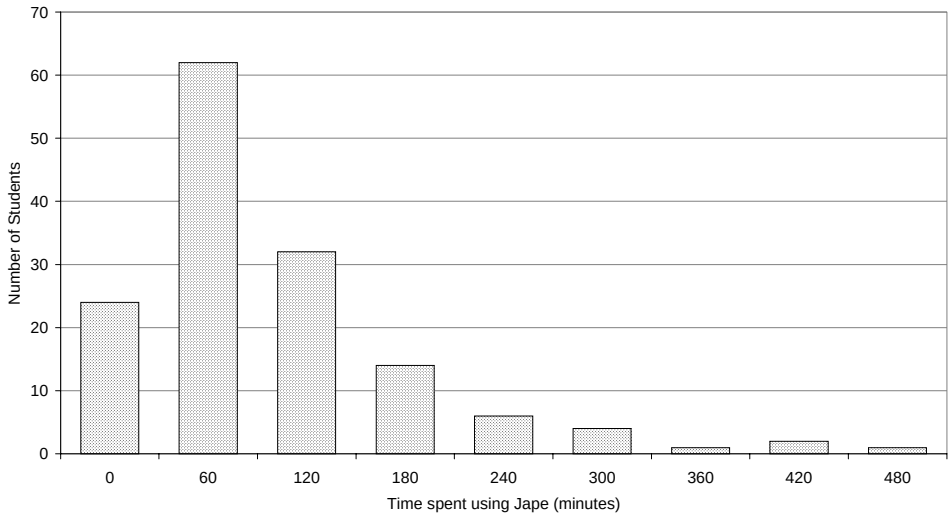


Figure 22: Frequencies of total time spent using Jape by students

The median time spent working on proofs was 59 minutes, but range was 4 minutes to 7.5 hours. On the whole, then, most students ended up using ItL Jape for no more than about two hours. However, since some did not use it at all (because of absence from those workshops) and some used it much more than most (because they accessed the program in their own time), there is a spread of usage time that will be useful in interpreting course outcome measures.

All but 5 students successfully completed at least one proof. About 70% of the proofs attempted were successfully completed. However, this varies widely from conjecture to conjecture and from student to student, as will be seen below.

The average proof attempt took almost 2½ minutes, successful proofs taking slightly less time and unsuccessful proofs taking slightly longer. Proofs were attempted at an average rate of 26 an hour; proofs were completed successfully at an average rate of 18 an hour. However, again, all these figures vary widely from conjecture to conjecture and from student to student.

All 70 conjectures built into ItL Jape were attempted at least once. Other than the 5 false conjectures, there were 3 conjectures that were not successfully proved by any of the students; and there were two conjectures that were proved by just one student each (different students). There were a further 13 conjectures that were successfully proved by fewer than 10 students.

4.4.6 Jape usage - by date

Number of students, by date

This table shows the pattern of Jape usage over the whole logic course.

Date	Number of Students
Mon 2 Nov 98	105
Tue 3 Nov 98	16
Wed 4 Nov 98	8
Thu 5 Nov 98	7
Fri 6 Nov 98	8
Sun 8 Nov 98	3
Mon 9 Nov 98	43
Tue 10 Nov 98	17
Wed 11 Nov 98	13
Thu 12 Nov 98	29
Fri 13 Nov 98	17
Sun 15 Nov 98	1
Mon 16 Nov 98	3
Wed 18 Nov 98	1
Fri 27 Nov 98	1
Mon 7 Dec 98	81
Tue 8 Dec 98	1
Wed 9 Dec 98	1
Thu 10 Dec 98	2
Fri 11 Dec 98	2
Mon 14 Dec 98	7
Tue 15 Dec 98	7
Wed 16 Dec 98	2
Thu 17 Dec 98	9
Fri 18 Dec 98	3
Tue 20 Apr 99	1
Wed 21 Apr 99	1
Thu 22 Apr 99	1
Fri 23 Apr 99	1
Tue 27 Apr 99	1
Thu 29 Apr 99	1
Fri 30 Apr 99	2
Thu 6 May 99	1
Fri 7 May 99	1
Fri 25 Jun 99	1
Tue 29 Jun 99	1
Thu 5 Aug 99	1
Thu 12 Aug 99	1
Sun 15 Aug 99	1
Wed 18 Aug 99	1

Figure 23: Number of students using Jape, by date

Note that this data underestimates Jape activity by about 20%, because there were about 30 students for whom logfiles are not available.

The table highlights the three main Jape workshops - 2 November, 9 November and 7 December. The diminished activity on 9 November compared to the other two workshops is because students had a choice on that date between attending a revision workshop and using Jape.

The pattern of activity between the second and third Jape workshops (9 November to 7 December) can be explained by noting that the first logic test was on 13 November. The table shows that some student chose to use Jape for revision in their own time just before this date, and that there was little usage afterwards.

The second logic test was on 18 December, and again there is activity just in the few days before this, although it appears to be much less than before the first test. The exam was on 17 May. The table above shows very little activity before this date.

Using this analysis, a new table can be constructed from the original data:

Time Period	Number of students
First Jape workshop	105
Between first two workshops	36
Second Jape workshop	43
Before Logic1 test	54
After Logic1 test	5
Third Jape workshop	81
Before Logic2 test	23
Before exam	7
After exam	2

Figure 24: Number of students using Jape, by time period

This table suggests that about a third of students who took Logic1 used Jape in their own time before the test (54 students). Only around 15% of students who took Logic2 used Jape in their own time before the test (23 students). Only 7 students (out of 145) used Jape in the time leading up to the exam. They were Charlotte (S19), Marion (S20), Angus (S34), Toby (S86), Philip (S92), Colin (S121), and Mary (S151).

It should be remembered that 10 students took part in the Reflection Study in the fortnight before the exam, and at least some of these students would have been tempted to have used Jape for themselves had this study not been in operation. Nevertheless, it is clear that very few students chose to use Jape for exam revision. There is a possible logistical reason - the campus was relatively empty at this time, as students revised at home and came in just for the exams. But even so, the value of Jape to the students was clearly not enough to persuade them that a trip into college to use the program was worthwhile.

Number of proof attempts, by time period

The following chart shows the number of proofs attempted, and the proportion of these that were successful.

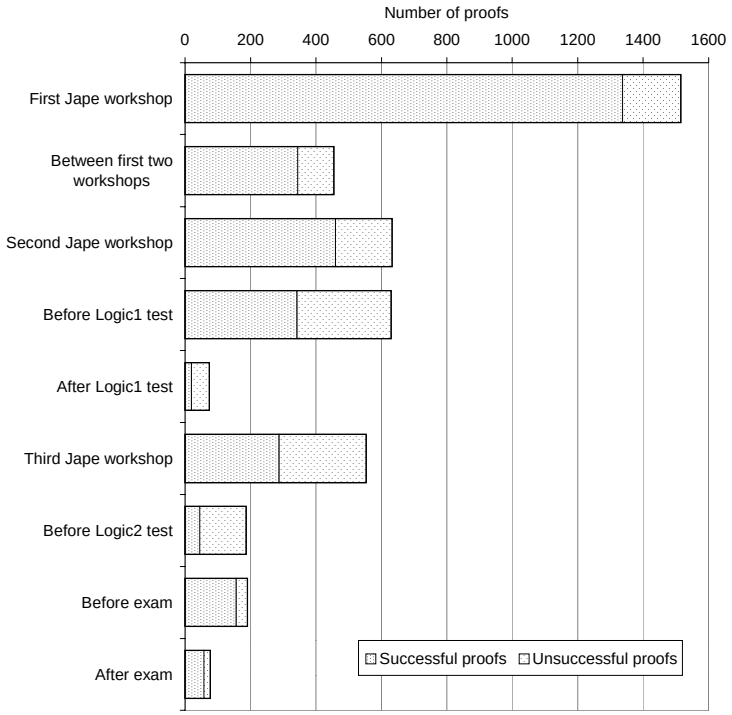


Figure 25: Total number of proofs tackled using Jape, by time period

The steady decline - over the three workshops - in the proportion of attempted proofs that were successful (from 88% down to 52%) is perhaps symptomatic of an increase in difficulty. Students reported that conjectures involving Negation and Quantifiers were harder than Implication, Conjunction and Disjunction.

This difficulty would appear not to be restricted to only a few students, as can be seen if the "intractable" conjectures are considered. These are the conjectures that none of the students could solve. In each workshop, around 40 conjectures were considered by the student cohort; yet while fewer than 15% of the conjectures attempted in either of the first two workshops were intractable, the figure for the third workshop is 68% (and this is excluding the user-entered and false conjectures that are present in this workshop and that would naturally increase the figure).

The period just before the exam would appear to have the greatest breadth of activity. Not only were *all* the 65 conjectures present in Jape attempted at this time (no more than 72% of the set were tackled in any of the other periods), but 60 of them were successfully completed by at least one student.

Time spent, by time period

The above identification of major dates of Jape activity by student numbers and by proof attempts is also reflected in the data for the number of hours spent using Jape. However, as the chart below shows, the decline shown in the number of proof attempts does not necessarily indicate a decline in Jape activity. (Note also that this chart does not include 333 proof attempts for which no time was recorded).

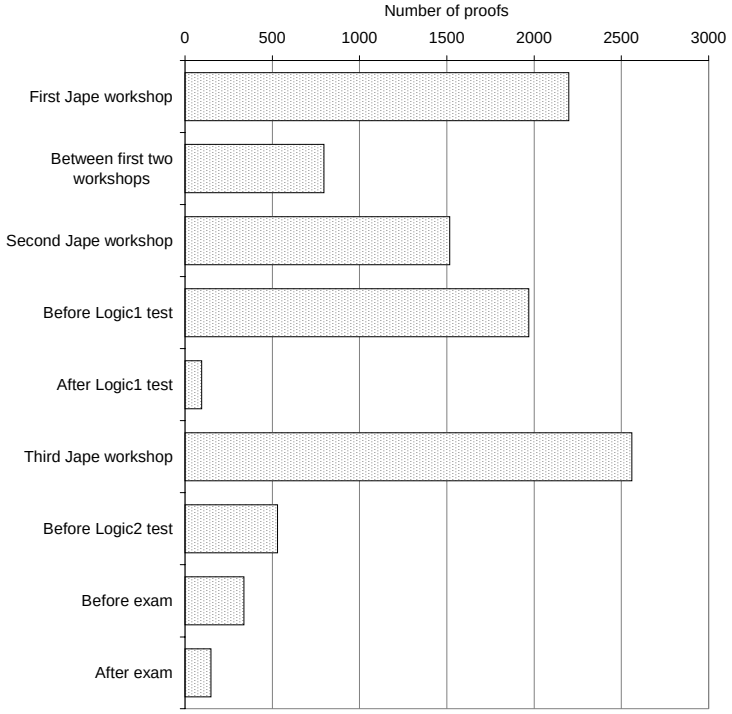


Figure 26: Time spent using Jape, by time period

The third Jape workshop shows fewer proof attempts than the other two workshops, yet this chart shows that more time was spent using Jape during this workshop than either of the other two. In the first Jape workshop, students spent on average 1½ minutes on each proof. In the second Jape workshop, this went up to almost 2½ minutes per proof. In the third Jape workshop, the average time spent on each proof was over 4½ minutes a proof. There is a corresponding decrease in success rate from 36 proofs an hour, to 18 proofs an hour, to 7 proofs an hour.

4.4.7 Jape usage - by conjecture

The appendix to this report contains detailed usage data for each conjecture, for example the number of students attempting, the number of successful students, the number of proof attempts, the proportion of proof attempts that are successful, the total time spent, and the average time spent per proof attempt. The order of the conjectures in the tables is the same as the order of the conjectures as presented to the students in ItL Jape.

Number of students attempting each conjecture

The following chart shows the steep decline in student attempts following the order in which the conjectures appear in Jape.

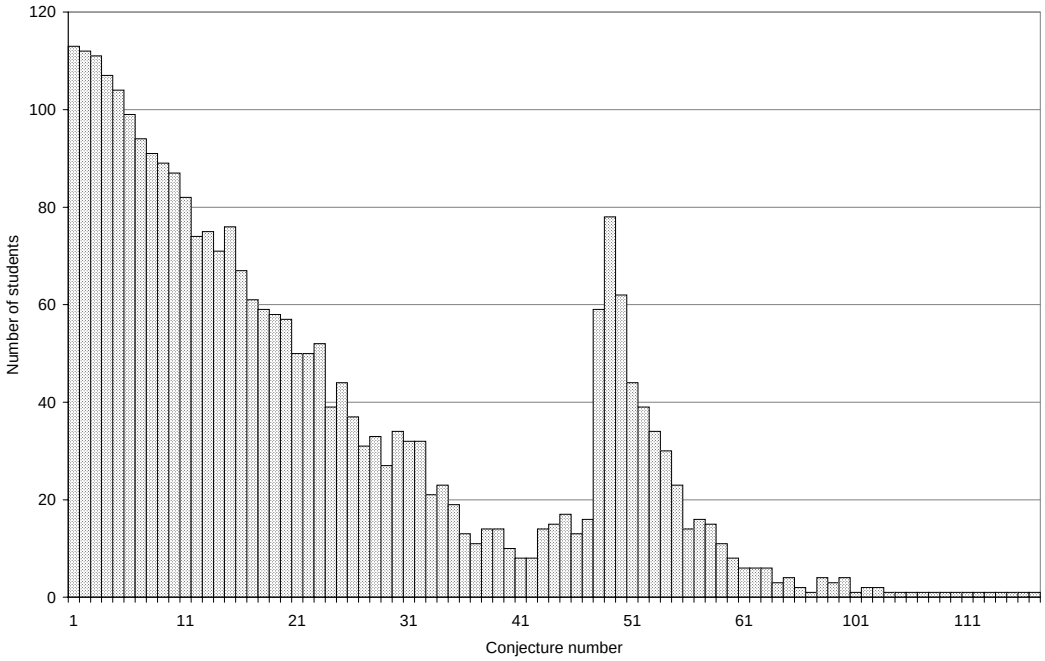


Figure 27: Number of students attempting each conjecture

The upturn at conjecture C48 - $\{\mathsf{var}\, c, P(c), \forall x.(P(x)\rightarrow Q(x)) \vdash Q(c)\}$ - is because this is the first quantifier proof; and in the third Jape workshop, students were encouraged to start with Quantifiers. It can be seen from the graph that over a third more students started at conjecture C49 - $\{\forall x.(P(x)\rightarrow Q(x)) \vdash \forall x.P(x)\rightarrow\forall x.Q(x)\}$ - than at C48.

Number of attempts at each conjecture

The variation in the number of attempts at each conjecture is similar to that of the number of students attempting each conjecture, described above.

Taking all the conjectures as a whole, there is an average of about 1.5 attempts per conjecture per student. Where there are deviations from this average, they tend to be for conjectures that have smaller numbers of students attempting them. There are some exceptions, however. For example, 33 students tackled conjecture 28 - $(P\vee Q)\wedge(P\vee R)\vdash P\vee(Q\wedge R)$ - with an average of 2.4 attempts per student. This anomaly is because four students attempted it repeatedly on different occasions before managing to prove it. Another anomaly is for conjecture 23 - $P\vee(Q\vee R)\vdash(P\vee Q)\vee R$ - which was tackled by 52 students with an average of 2.3 attempts per student. This conjecture has the highest number of unsuccessful attempts of the conjectures in the Disjunction topic. One further anomaly is conjecture C49 - $\forall x.(P(x)\rightarrow Q(x))\vdash\forall x.P(x)\rightarrow\forall x.Q(x)$ - which was tackled by 78 students with an average of 2.0 attempts per student. This conjecture has the highest number of unsuccessful attempts out of all the conjectures included in ItL Jape. The Conjunction conjecture with the highest number of unsuccessful attempts was $(P\wedge Q)\rightarrow R\vdash P\rightarrow(Q\rightarrow R)$ (C15). For the Implication topic, the conjectures $P\rightarrow(Q\rightarrow P)$ (C8) and $(P\rightarrow(Q\rightarrow R))\rightarrow((P\rightarrow Q)\rightarrow(P\rightarrow R))$ (C11) have the highest number of unsuccessful attempts.

There is wide variation, from conjecture to conjecture, in the proportion of successful proof attempts:

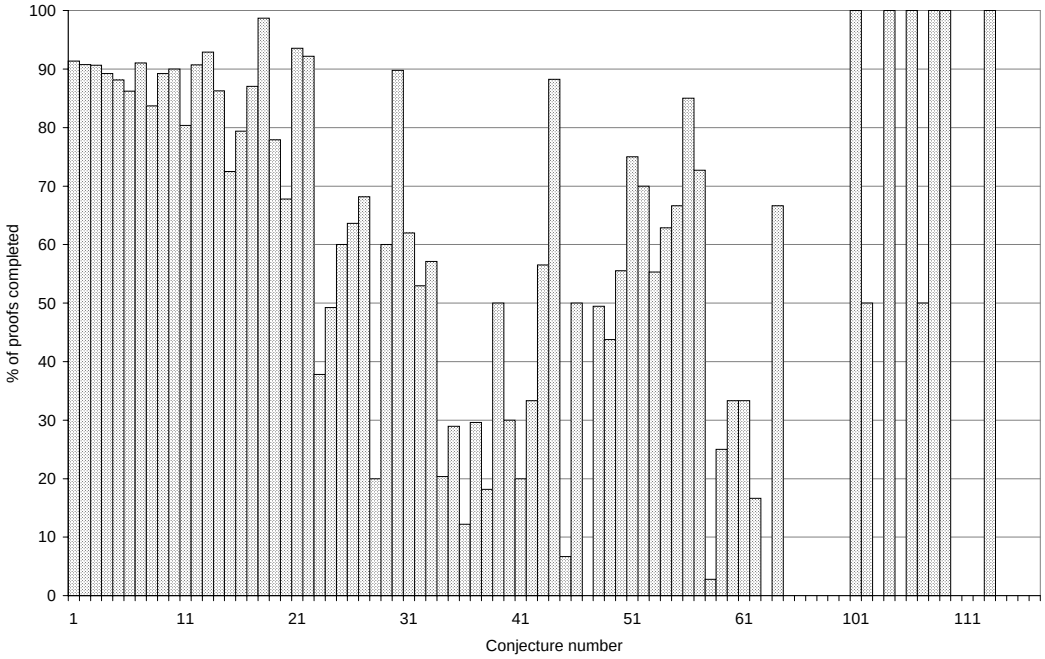


Figure 28: Proportion of proof attempts that were successful, by conjecture

Note that conjectures 101-117 were entered by the students, so some of the conjectures may be false, and some of the proofs may be trivial. In any case, the number of proof attempts is small for each of these conjectures.

Conjectures 66-70 are false (and are labelled as such).

There are 3 conjectures that were unproven by any student:

Conjecture Number	Conjecture	Number of attempts	Number of students attempting
47	$((P \rightarrow Q) \rightarrow P) \rightarrow P$	28	16
63	$\exists x.\neg P(x) \vdash \neg(\forall x.P(x))$	7	6
65	$\forall x.\neg P(x) \vdash \neg(\exists x.P(x))$	4	4

Figure 29: Conjectures that no students proved

The following table shows the "least popular conjectures": that is, those conjectures that were attempted by fewer than 20 students, excluding the user-entered and false conjectures:

Conjecture Number	Conjecture	Topic	Number of students attempting	Number of successful students	Number of proof attempts	Number of successful proof attempts	Time spent on proof attempts (mins)	Time spent on successful proofs (mins)	Time per proof attempt (mins)	Time per successful proof (mins)	Time per student on proof attempts (mins)	Time per student on successful proofs (mins)
64	$\neg(\exists x.P(x)) \vdash \forall x.\neg P(x)$	Quantifiers	3	2	3	2	3	3	0.9	1.3	0.9	0.9
65	$\forall x.\neg P(x) \vdash \neg(\exists x.P(x))$	Quantifiers	4	0	4	0	12	0	3.0		3.0	0.0
61	$\neg(\forall x.\neg P(x)) \vdash \exists x.P(x)$	Quantifiers	6	2	6	2	17	6	2.9	3.2	2.9	1.1
62	$\neg(\forall x.P(x)) \vdash \exists x.\neg P(x)$	Quantifiers	6	1	6	1	11	2	1.8	2.3	1.8	0.4
63	$\exists x.\neg P(x) \vdash \neg(\forall x.P(x))$	Quantifiers	6	0	7	0	8	0	1.2		1.4	0.0
41	$\neg(P \wedge Q) \vdash \neg P \vee \neg Q$	Negation	8	3	15	3	43	7	2.9	2.4	5.4	0.9
42	$\neg P \vee \neg Q \vdash \neg(P \wedge Q)$	Negation	8	5	18	6	50	18	2.8	3.1	6.3	2.3
60	$\exists x.P(x) \vdash \neg(\forall x.\neg P(x))$	Quantifiers	8	2	9	3	16	3	1.8	1.1	2.0	0.4
40	$\neg P \wedge \neg Q \vdash \neg(P \vee Q)$	Negation	10	4	20	6	53	27	2.7	4.4	5.3	2.7
37	$P \wedge Q \vdash \neg(\neg P \vee \neg Q)$	Negation	11	6	27	8	110	53	4.1	6.7	10.0	4.9
59	$\neg(\exists x.\neg P(x)) \vdash \forall x.P(x)$	Quantifiers	11	3	12	3	35	4	2.9	1.3	3.2	0.3
36	$\neg(\neg P \wedge \neg Q) \vdash P \vee Q$	Negation	13	5	41	5	147	73	3.6	14.7	11.3	5.6
46	$P \wedge \neg P \vdash Q$	Negation	13	6	16	8	17	11	1.1	1.4	1.3	0.9
38	$\neg(\neg P \vee \neg Q) \vdash P \wedge Q$	Negation	14	3	22	4	34	13	1.5	3.3	2.4	1.0
39	$\neg(P \vee Q) \vdash \neg P \wedge \neg Q$	Negation	14	8	18	9	71	34	3.9	3.8	5.0	2.4
43	$\neg(P \wedge \neg P)$	Negation	14	9	23	13	18	12	0.8	0.9	1.3	0.8
56	$\exists x.(P(x) \vee Q(x)) \vdash \exists x.P(x) \vee \exists x.Q(x)$	Quantifiers	14	14	20	17	83	82	4.1	4.8	5.9	5.9
44	$Q \rightarrow P, P \rightarrow R \vdash Q \rightarrow R$	Negation	15	13	17	15	13	10	0.8	0.7	0.9	0.7
58	$\forall x.P(x) \vdash \neg(\exists x.\neg P(x))$	Quantifiers	15	1	36	1	83	5	2.3	4.7	5.5	0.3
47	$((P \rightarrow Q) \rightarrow P) \rightarrow P$	Negation	16	0	28	0	113	0	4.0		7.1	0.0
57	$\text{var } c, \forall x.P(x) \vdash \exists x.P(x)$	Quantifiers	16	14	22	16	70	34	3.2	2.1	4.4	2.1
45	$(P \rightarrow Q) \vee (Q \rightarrow P)$	Negation	17	2	30	2	123	16	4.1	8.1	7.2	0.9
35	$P \vee Q \vdash \neg(\neg P \wedge \neg Q)$	Negation	19	8	38	11	103	36	2.7	3.3	5.4	1.9

Figure 30: The least popular conjectures

It will be noticed that all the least popular conjectures are from the Negation and Quantifier topics. It is also interesting that, on average, the proportion of students who attempted these conjectures who were successful was 40%, compared to 83% for the popular conjectures (the conjectures each of which were attempted by more than 20 students). This distinction is similar for the proportion of proof attempts that were successful and for the proportion of time that was spent on successful proofs. There is little difference, however, between the popular and unpopular conjectures as regards the average number of proof attempts per student, or as regards the average time per proof attempt. However, *successful* proof attempts for the unpopular conjectures took around 40 seconds longer on average than for the popular conjectures (equivalent to about 25% longer).

The "facility" of a conjecture could be measured in several ways: % of the attempts at the conjecture that are successfully completed; % of the students who attempted the conjecture who are successful at least once; % of the

time spent tackling conjectures that was spent on ultimately successful conjectures. All these measures are highly correlated, although there are subtle differences between them.

The following table shows the "popular conjectures" (the conjectures each of which were attempted by more than 20 students), ranked in ascending order by the proportion of the students who attempted the conjecture who were successful at least once.

Conjecture Number	Conjecture	Topic	Number of students attempting	Number of successful students	% of attempting students who are successful	Number of proof attempts	Number of successful proof attempts	% of proof attempts that are successful	Number of proof attempts per student	Time spent on proof attempts (mins)	Time spent on successful proofs (mins)	% of time that is spent on successful proofs	Time per proof attempt (mins)	Time per successful proof (mins)	Time per student on proof attempts (mins)	Time per student on successful proofs (mins)
34	$P \vee \neg P$	Negation	23	8	35	54	11	20	2.3	232	107	46	4.3	9.7	10.1	4.6
28	$(P \vee Q) \wedge (P \vee R) \vdash P \vee (Q \wedge R)$	Disjunction	33	12	36	80	16	20	2.4	330	109	33	4.1	6.8	10.0	3.3
33	$\neg Q \rightarrow \neg P \vdash P \rightarrow Q$	Negation	21	13	62	35	20	57	1.7	103	73	71	3.0	3.6	4.9	3.5
32	$P \rightarrow Q \vdash \neg Q \rightarrow \neg P$	Negation	32	20	63	51	27	53	1.6	170	84	49	3.3	3.1	5.3	2.6
48	$\text{var } c, P(c), \forall x.(P(x) \rightarrow Q(x)) \vdash Q(c)$	Quantifiers	59	38	64	95	47	49	1.6	323	215	67	3.4	4.6	5.5	3.6
55	$\exists x.P(x) \vee \exists x.Q(x) \vdash \exists x.(P(x) \vee Q(x))$	Quantifiers	23	15	65	24	16	67	1.0	99	89	90	4.1	5.6	4.3	3.9
54	$\exists x.(P(x) \wedge Q(x)) \vdash \exists x.P(x) \wedge \exists x.Q(x)$	Quantifiers	30	20	67	35	22	63	1.2	125	69	56	3.6	3.2	4.2	2.3
23	$P \vee (Q \vee R) \vdash (P \vee Q) \vee R$	Disjunction	52	35	67	119	45	38	2.3	638	244	38	5.4	5.4	12.3	4.7
29	$P \rightarrow R, Q \rightarrow R \vdash (P \vee Q) \rightarrow R$	Disjunction	27	19	70	40	24	60	1.5	98	56	57	2.4	2.3	3.6	2.1
53	$\forall x.(P(x) \rightarrow Q(x)), \exists x.P(x) \vdash \exists x.Q(x)$	Quantifiers	34	24	71	47	26	55	1.4	179	131	73	3.8	5.0	5.3	3.8
24	$(P \vee Q) \vee R \vdash P \vee (Q \vee R)$	Disjunction	39	28	72	65	32	49	1.7	236	93	40	3.6	2.9	6.0	2.4
49	$\forall x.(P(x) \rightarrow Q(x)) \vdash \forall x.P(x) \rightarrow \forall x.Q(x)$	Quantifiers	78	58	74	153	67	44	2.0	1076	812	75	7.0	12.1	13.8	10.4
25	$P \wedge (Q \vee R) \vdash (P \wedge Q) \vee (P \wedge R)$	Disjunction	44	33	75	65	39	60	1.5	322	175	54	5.0	4.5	7.3	4.0
31	$P \vdash \neg \neg P$	Negation	32	24	75	50	31	62	1.6	96	60	63	1.9	1.9	3.0	1.9
51	$\forall x.P(x) \wedge \forall x.Q(x) \vdash \forall x.(P(x) \wedge Q(x))$	Quantifiers	44	34	77	52	39	75	1.2	151	118	78	2.9	3.0	3.4	2.7
26	$(P \wedge Q) \vee (P \wedge R) \vdash P \wedge (Q \vee R)$	Disjunction	37	29	78	55	35	64	1.5	215	170	79	3.9	4.8	5.8	4.6
52	$\forall x.(P(x) \wedge Q(x)) \vdash \forall x.P(x) \wedge \forall x.Q(x)$	Quantifiers	39	31	79	50	35	70	1.3	332	301	91	6.6	8.6	8.5	7.7
50	$\forall x.(P(x) \rightarrow Q(x)), \forall x.(Q(x) \rightarrow R(x)) \vdash \forall x.(P(x) \rightarrow R(x))$	Quantifiers	62	51	82	99	55	56	1.6	364	291	80	3.7	5.3	5.9	4.7
27	$P \vee (Q \wedge R) \vdash (P \vee Q) \wedge (P \vee R)$	Disjunction	31	26	84	44	30	68	1.4	152	124	82	3.4	4.1	4.9	4.0
20	$Q \rightarrow R \vdash (P \vee Q) \rightarrow (P \vee R)$	Disjunction	57	49	86	87	59	68	1.5	377	209	55	4.3	3.5	6.6	3.7
15	$(P \wedge Q) \rightarrow R \vdash P \rightarrow (Q \rightarrow R)$	Conjunction	76	67	88	120	87	73	1.6	311	224	72	2.6	2.6	4.1	2.9
16	$P \rightarrow (Q \rightarrow R) \vdash (P \wedge Q) \rightarrow R$	Conjunction	67	60	90	97	77	79	1.4	210	173	83	2.2	2.2	3.1	2.6
11	$(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$	Implication	82	74	90	112	90	80	1.4	161	142	88	1.4	1.6	2.0	1.7
30	$\neg \neg P \rightarrow P$	Negation	34	31	91	49	44	90	1.4	51	48	94	1.0	1.1	1.5	1.4
9	$P \rightarrow Q \vdash (Q \rightarrow R) \rightarrow (P \rightarrow R)$	Implication	89	84	94	121	108	89	1.4	143	115	81	1.2	1.1	1.6	1.3
19	$P \vee Q \vdash Q \vee P$	Disjunction	58	55	95	86	67	78	1.5	242	168	69	2.8	2.5	4.2	2.9
6	$P \rightarrow (Q \rightarrow R) \vdash (P \rightarrow Q) \rightarrow (P \rightarrow R)$	Implication	99	94	95	138	119	86	1.4	308	286	93	2.2	2.4	3.1	2.9
17	$P \vdash P \vee Q$	Disjunction	61	58	95	85	74	87	1.4	53	47	88	0.6	0.6	0.9	0.8
21	$P \vee P \vdash P$	Disjunction	50	48	96	62	58	94	1.2	37	31	84	0.6	0.5	0.7	0.6
5	$P \rightarrow (Q \rightarrow R) \vdash Q \rightarrow (P \rightarrow R)$	Implication	104	100	96	152	134	88	1.5	340	317	93	2.2	2.4	3.3	3.0
3	$P \rightarrow (Q \rightarrow R), P \rightarrow Q, P \vdash R$	Implication	111	107	96	161	146	91	1.5	188	184	98	1.2	1.3	1.7	1.7
8	$P \rightarrow (Q \rightarrow P)$	Implication	91	88	97	135	113	84	1.5	71	57	79	0.5	0.5	0.8	0.6
14	$P \wedge Q \vdash Q$	Conjunction	71	69	97	102	88	86	1.4	38	26	69	0.4	0.3	0.5	0.4
13	$P \wedge Q \vdash P$	Conjunction	75	73	97	99	92	93	1.3	70	68	97	0.7	0.7	0.9	0.9
10	$P \rightarrow (Q \rightarrow (R \rightarrow S)) \vdash R \rightarrow (Q \rightarrow (P \rightarrow S))$	Implication	87	85	98	120	108	90	1.4	226	208	92	1.9	1.9	2.6	2.4
22	$P \vdash P \vee P$	Disjunction	50	49	98	64	59	92	1.3	18	18	97	0.3	0.3	0.4	0.4
2	$P \rightarrow Q, Q \rightarrow R, P \vdash R$	Implication	112	110	98	173	157	91	1.5	117	112	96	0.7	0.7	1.0	1.0
1	$P, P \rightarrow Q \vdash Q$	Implication	113	111	98	197	180	91	1.7	112	85	76	0.6	0.5	1.0	0.8
12	$P, Q \vdash P \wedge Q$	Conjunction	74	73	99	97	88	91	1.3	61	50	82	0.6	0.6	0.8	0.7
4	$P \rightarrow Q, Q \rightarrow R \vdash P \rightarrow R$	Implication	107	107	100	158	141	89	1.5	334	292	87	2.1	2.1	3.1	2.7
18	$Q \vdash P \vee Q$	Disjunction	59	59	100	77	76	99	1.3	52	52	99	0.7	0.7	0.9	0.9
7	$P \vdash Q \rightarrow P$	Implication	94	94	100	134	122	91	1.4	36	31	86	0.3	0.3	0.4	0.3

Figure 31: The popular conjectures

Time spent on each conjecture

The following chart suggests that in the early stages, the time spent on each proof seems to depend on the textual length of the statement of the conjecture.

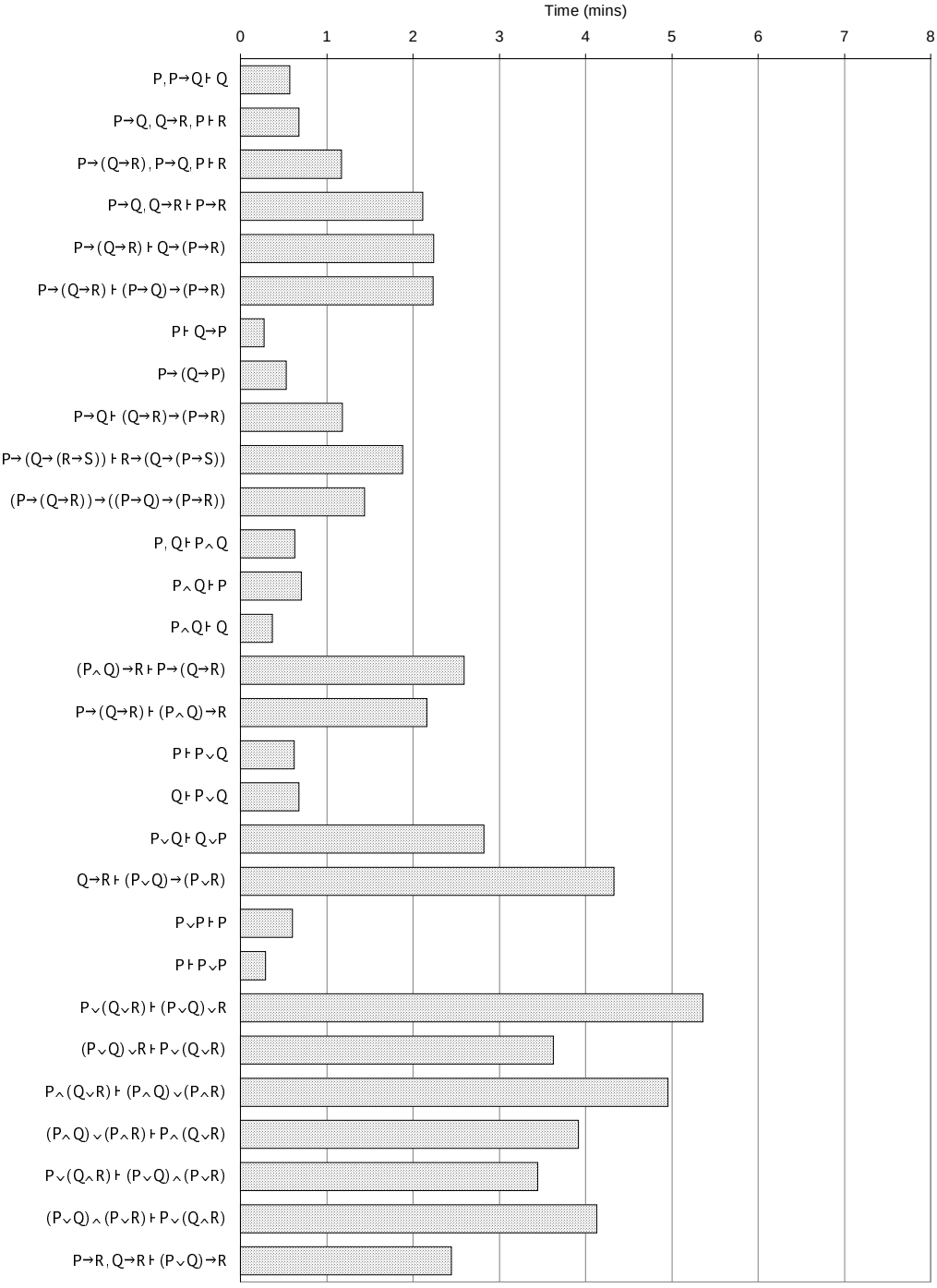


Figure 32: Time spent per attempt on each conjecture, for Implication, Conjunction & Disjunction (conjectures 1-29)

The correlation between textual length and time taken is significant: Spearman's rho for conjectures attempted by more than 50 students is 0.78 ($p<0.001$).

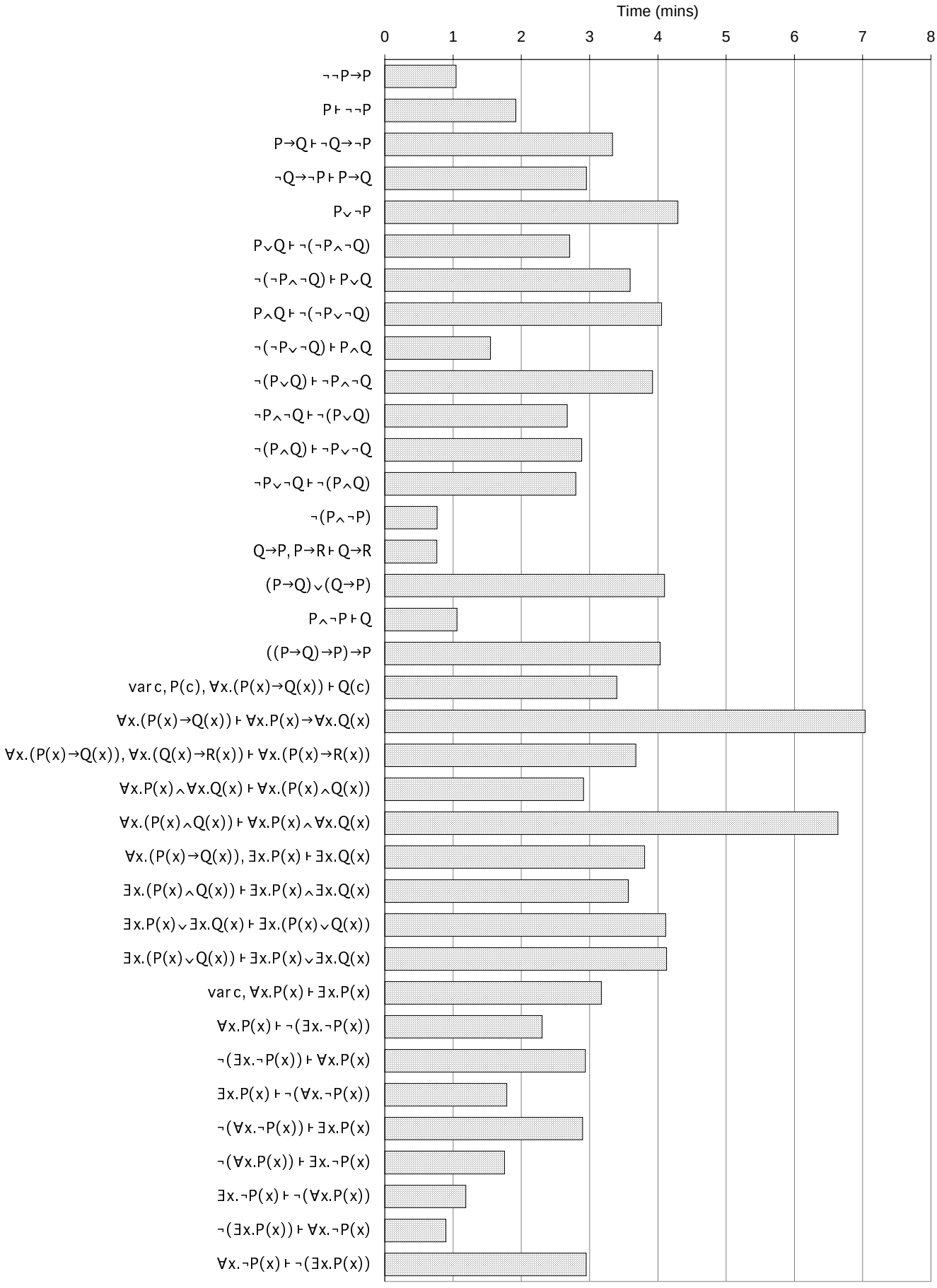


Figure 33: Time spent per attempt on each conjecture, for Negation & Quantifiers (conjectures 30-65)

Conjecture 34 - P∨¬P - is the first strong counter-example to the simple characterisation of conjecture difficulty in terms of the length of the statement. This conjecture also has the highest number of unsuccessful attempts of the conjectures in the Negation topic.

The following charts show, for conjectures with more than 20 failed attempts, the average time before a proof attempt is abandoned:

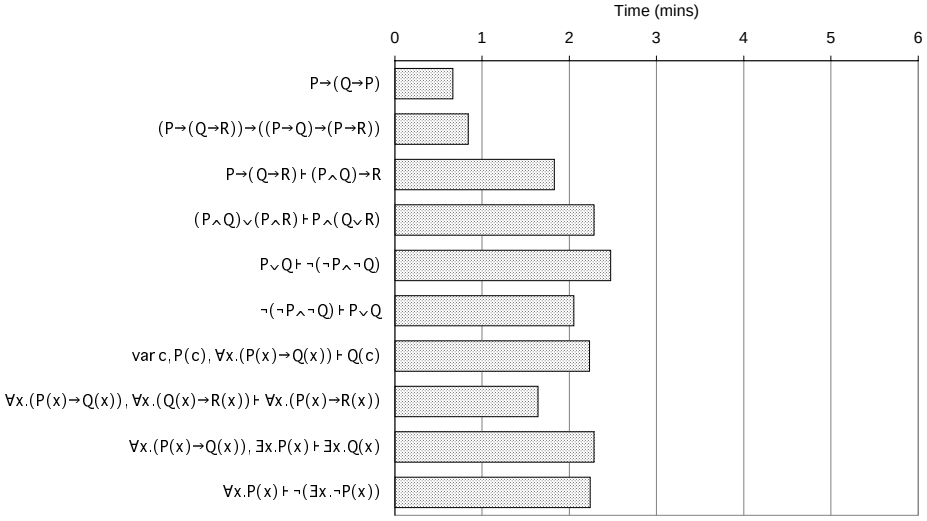


Figure 34: Conjectures that were abandoned quickly

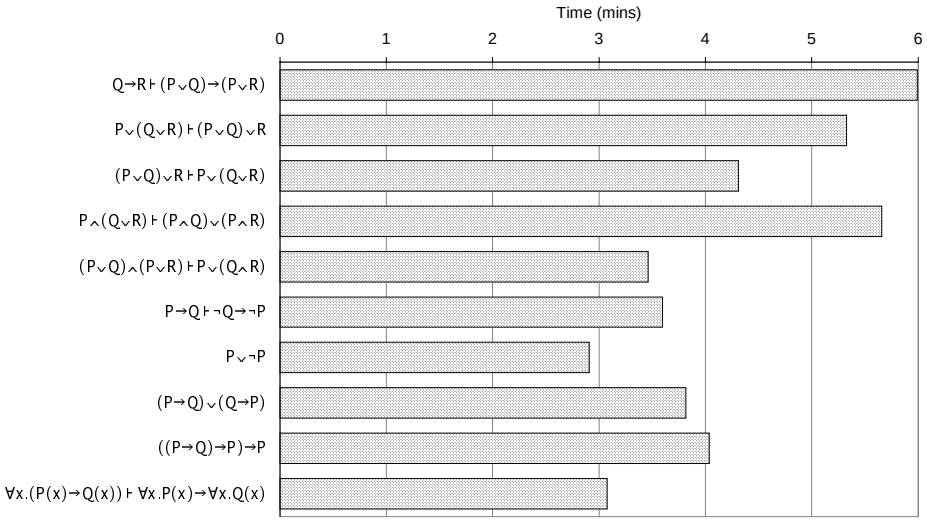


Figure 35: Conjectures that were abandoned slowly

Of course, it should be noted that students may abandon proofs for a variety of reasons other than giving up. For example, they just may not have the time available to proof it (at the end of a workshop, say); or they may already have proved this conjecture in another window. But what is it about the display of certain conjectures - in relation to student's previous experiences - that makes them likely to be abandoned more quickly than others?

This chart shows the strong relationship (Pearson = -0.72) between the proportion of students who prove a particular conjecture at least once, and the time they take to prove that conjecture.

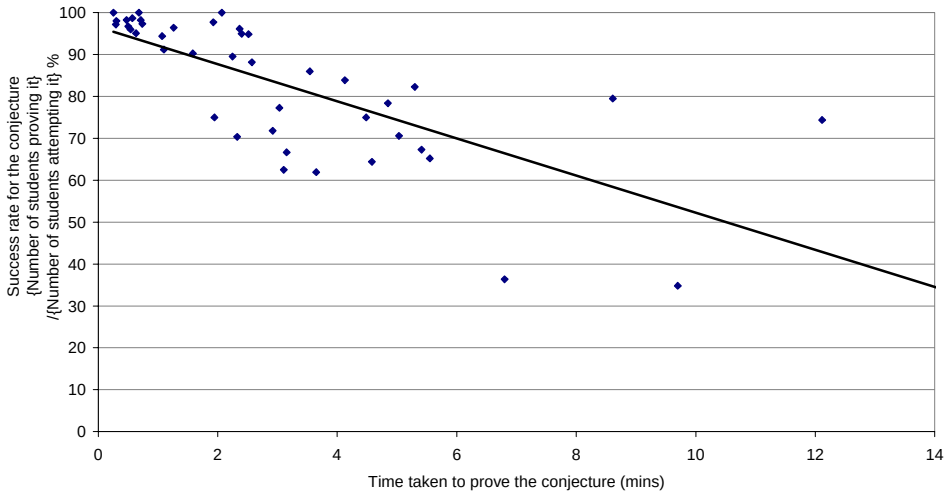


Figure 36: For each conjecture tackled by more than 20 students, comparison of the success rate of the conjecture and the average duration of the conjecture

Minimum time to prove each conjecture

Judging by the minimum times taken by any student to prove each conjecture, each of the 11 conjectures in the Implication topic could in principle be proved within 20 seconds. The whole topic, indeed, could in principle be proved within 90 seconds. In reality, of course, selecting the conjecture to prove and indicating that it has been proved take some extra time. The most time-consuming conjecture in the Implication topic was $P \rightarrow (Q \rightarrow (R \rightarrow S)) \vdash R \rightarrow (Q \rightarrow (P \rightarrow S))$ (C10), which Colin (S121) proved in 16 seconds.

Each of the 5 conjectures in the Conjunction topic could also in principle be proved within about 20 seconds. The whole topic could in principle be proved in 45 seconds. The most time-consuming conjecture in the Conjunction topic was $P \rightarrow (Q \rightarrow R) \vdash (P \wedge Q) \rightarrow R$ (C16), which Helen (S105) proved in 21 seconds.

The 13 conjectures in the Disjunction topic are generally more time-consuming than those in the Implication or Conjunction topics. The shortest time to prove the topic would, on these figures, be 7 minutes. Just 4 conjectures could be proved in under 20 seconds; and 8 within one minute. The most time-consuming conjecture in the Disjunction topic was $(P \vee Q) \wedge (P \vee R) \vdash P \vee (Q \wedge R)$ (C28), which Philip (S92) proved in 87 seconds.

The 18 conjectures in the Negation topic were at least as time-consuming as the Disjunction topic. The shortest time to prove the topic would be 16 minutes, however this does not include $((P \rightarrow Q) \rightarrow P) \rightarrow P$ (C46), which no-one proved. Six conjectures could be proved within 20 seconds, including $\neg\neg P \rightarrow P$ (C30), which Randolph (S137) proved in just 1 second. A further 5 could be proved within a minute; and 6 could take over a minute. Nevertheless, far fewer students attempted this topic than the previous topic. Indeed, just 5 students attempted $\neg(\neg P \wedge \neg Q) \vdash P \vee Q$ (C36), which Clement (S40) proved in just over 3 minutes.

The 18 conjectures in the Quantifiers topic were generally the most time-consuming of all. The shortest time to prove the topic would be 19 minutes, however this does not include either $\exists x.\neg P(x) \vdash \neg(\forall x.P(x))$ (C63) or $\forall x.\neg P(x) \vdash \neg(\exists x.P(x))$ (C65) which no-one proved. No conjectures could be proved within 20 seconds. The least time-consuming conjecture was the first one $\mathrm{var}\, c, P(c), \forall x.(P(x) \rightarrow Q(x)) \vdash Q(c)$ (C48), which Alan (S118) proved in 23 seconds. 11 conjectures could be proved within a minute; and 5 conjectures could be proved in over a minute. Just one student attempted $\forall x.P(x) \vdash \neg(\exists x.\neg P(x))$ (C58) - Keiko (S147) - and it took her almost 5 minutes.

In principle, then, the shortest time to prove all the true conjectures in ItL Jape (with the exception of the three that no-one proved) would be 45 minutes.

Order in which conjectures were attempted

Did students generally attempt conjectures in the order in which they were presented in ItL Jape? The answer is yes: out of all the proof attempts that were on a conjecture different from that of the previous proof attempt by that student on that day, 78% were on the next conjecture in the sequence of ItL Jape conjectures; 11% were on a later conjecture in the sequence; and just 10% were from a previous conjecture in the sequence.

Key conjectures

Assuming that students generally attempted conjectures in the order in which they were presented in ItL Jape (see the previous section), one can look at the decline in the number of students attempting each subsequent conjecture.

For example, for conjecture C11 - $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$ - there is a 10% drop in the number of students attempting the next conjecture, from 82 to 74. This could be related to the 7% drop in the proportion of attempting students who got this conjecture compared to the previous conjecture. Of the 74 students who proved C11, all but one also proved the other 10 Implication conjectures; and 77% proved all the Conjunction conjectures too. Of the 51 students who didn't prove C11 (either because they didn't attempt the conjecture, or because they attempted the conjecture and failed to prove it), just 9 proved the remaining 10 Implication conjectures; and just one student proved all the Conjunction conjectures.

Conjecture C15 - $(P \wedge Q) \rightarrow R \vdash P \rightarrow (Q \rightarrow R)$ - has a 12% drop in the number of students attempting the next conjecture, from 76 to 67. There is a 9% drop in the proportion of attempting students who got this conjecture compared to the previous conjecture.

Conjecture C23 - $P \vee (Q \vee R) \vdash (P \vee Q) \vee R$ - has a 25% drop in the number of students attempting the next conjecture, from 52 to 39. There is a 31% drop in the proportion of attempting students who got this conjecture compared to the previous conjecture. Of the 35 students who proved C23, 33 had proved all of Implication and 32 had proved all of Conjunction.

Conjecture C28 - $(P \vee Q) \wedge (P \vee R) \vdash P \vee (Q \wedge R)$ - has an 18% drop in the number of students attempting the next conjecture, from 33 to 27. There is a 48% drop in the proportion of attempting students who got this conjecture compared to the previous conjecture, from 84% to 36%. Of the 12 students who proved C28, 11 had proved all of Implication, 10 had proved all of Conjunction, and 10 went on to prove all of Disjunction.

Conjecture C32 - $P \rightarrow Q \vdash \neg Q \rightarrow \neg P$ - has an 34% drop in the number of students attempting the next conjecture, from 32 to 21. There is, however, only a 13% drop in the proportion of attempting students who got this conjecture compared to the previous conjecture.

Then there is a dramatic rise - already noted - in the number of attempting students, because of the third Jape workshop starting at Quantifiers.

Conjecture C49 - $\forall x.(P(x) \rightarrow Q(x)) \vdash \forall x.P(x) \rightarrow \forall x.Q(x)$ - has a 21% drop in the number of students attempting the next conjecture, from 78 to 62. The proportion of attempting students who got this conjecture compared to the previous conjecture actually rises by 10%. Success in this conjecture is not an especially good indicator of success in the previous topics.

Conjecture C50 - $\forall x.(P(x) \rightarrow Q(x)) - \forall x.(Q(x) \rightarrow R(x)) \vdash \forall x.(P(x) \rightarrow R(x))$ - has a 29% drop in the number of students attempting the next conjecture, from 62 to 44. The proportion of attempting students who got this conjecture compared to the previous conjecture again rises, this time by 8%. Although success in this conjecture is again not an especially good indicator of success in the previous topics, of the 74 students who did not proved this conjecture, just 4 had proved more than a couple of Negation conjectures.

Conjecture C54 - $\exists x.(P(x) \wedge Q(x)) \vdash \exists x.P(x) \wedge \exists x.Q(x)$ - has a 23% drop in the number of students attempting the next conjecture, from 30 to 23. There a negligible change in the proportion of attempting students who got this conjecture compared to the previous conjecture. Of the 20 students who proved C54, 19 had proved all of Implication and 17 had proved all of Conjunction.

Conjecture C55 - $\exists x.P(x) \vee \exists x.Q(x) \vdash \exists x.(P(x) \vee Q(x))$ - has a 39% drop in the number of students attempting the next conjecture, from 23 to 14. There is a negligible change in the proportion of attempting students who got this conjecture compared to the previous conjecture. Of the 15 students who proved C55, all but one had proved all of Implication and Conjunction.

Another way of looking at key conjectures is to see which ones tended to be final attempts for students.

For example, of the 37 students who missed out on proving C8 - $P \rightarrow (Q \rightarrow P)$ - just 1 went on to prove anything in Conjunction, 3 went on to prove anything in Disjunction and 5 went on to prove anything in Negation. Conversely, of the 88 students who proved C8, 75 (85%) proved at least one of the Conjunction conjectures, 63 (72%) proved at least one of the Disjunction conjectures, and 36 (41%) proved at least one of the Negation conjectures.

The Implication conjecture C10 - $P \rightarrow (Q \rightarrow (R \rightarrow S)) \vdash R \rightarrow (Q \rightarrow (P \rightarrow S))$ - is another key conjecture for Conjunction, Disjunction and Negation in a similar way.

The Conjunction conjecture C12 - $P, Q \vdash P \wedge Q$ - is another key conjecture for Disjunction. Of the 52 students who missed out on proving it, just 5 went on to prove anything in Disjunction. Conversely, of the 73 students who proved C12, 61 (84%) proved at least one of the Disjunction conjectures. The Conjunction conjecture C13 - $P \wedge Q \vdash P$ - is another key conjecture for Disjunction in a similar way.

The Quantifiers conjecture C49 - $\forall x.(P(x) \rightarrow Q(x)) \vdash \forall x.P(x) \rightarrow \forall x.Q(x)$ - is a key conjecture for Quantifiers. Of the 67 students who missed out on proving it, just 6 proved anything else in Quantifiers. Conversely, of the 58 students who proved C49, 53 (91%) proved at least one other Quantifier conjecture.

Another way of looking at key conjectures is to rank the usage figures to see which conjectures might have caused particular problems. There are four conjectures which regularly feature in the lists of conjectures - attempted by at least 20 students - that have the highest number of unsuccessful proof attempts, the highest proportion of unsuccessful proof attempts, the highest number of proof attempts per student, the longest time spent, the longest time spent per student, and so on:

Conjecture number	Conjecture
23	$P \vee (Q \vee R) \vdash (P \vee Q) \vee R$
28	$(P \vee Q) \wedge (P \vee R) \vdash P \vee (Q \wedge R)$
34	$P \vee \neg P$
49	$\forall x.(P(x) \rightarrow Q(x)) \vdash \forall x.P(x) \rightarrow \forall x.Q(x)$

Figure 37: Conjectures causing difficulties

4.4.8 Jape usage - by duration of proof attempts

How long proof attempts lasted

This chart excludes 333 proof attempts for which no times were recorded.

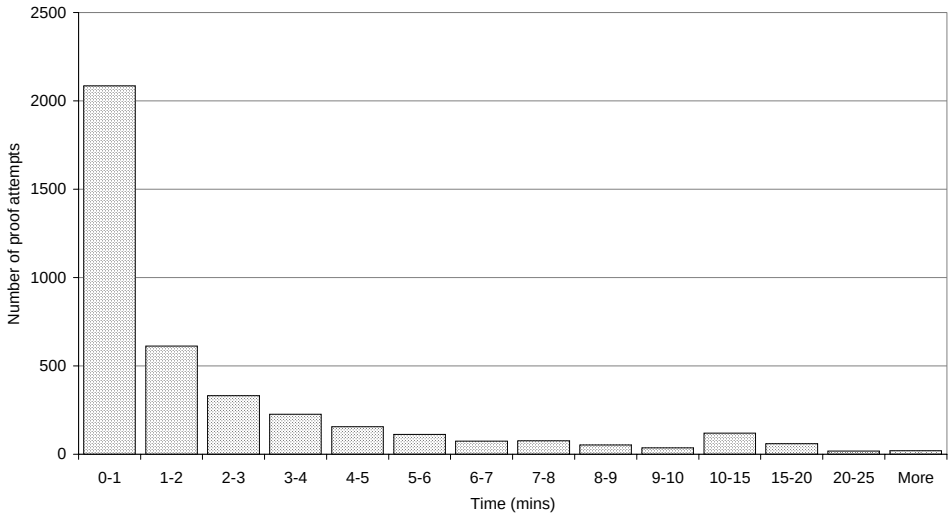


Figure 38: Time spent on proof attempts

Over half of all proof attempts lasted no more than a minute, and three-quarters lasted no more than 3 minutes.

The charts below show how long students spent on the "difficult" conjectures identified in Figure 37. In each case, the proof attempts for which no times were recorded have been excluded. Note too that, because these conjectures feature high in the list of conjectures with a large number of proof attempts per student, the charts showing the times for *students* (as opposed to proof attempts) would be skewed further to the right than these charts.

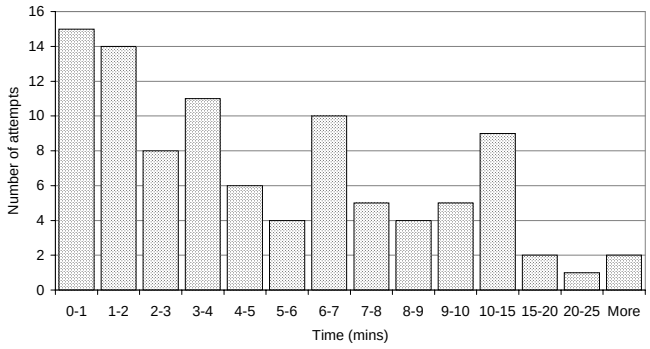


Figure 39: Time spent on attempts at conjecture C23

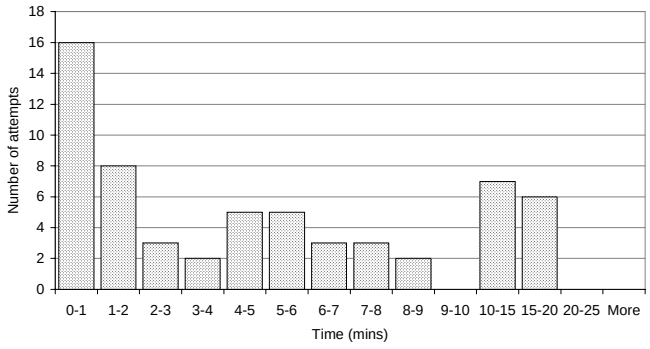


Figure 40: Time spent on attempts at conjecture C28

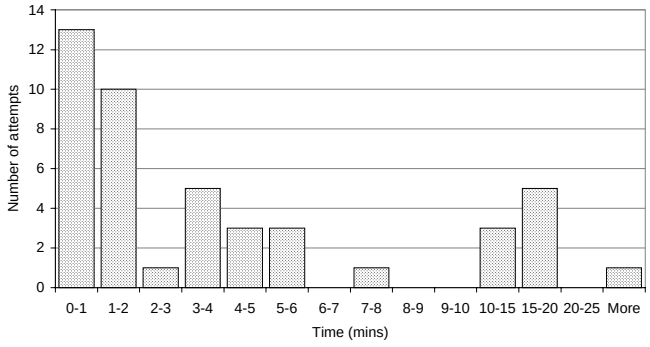


Figure 41: Time spent on attempts at conjecture C34

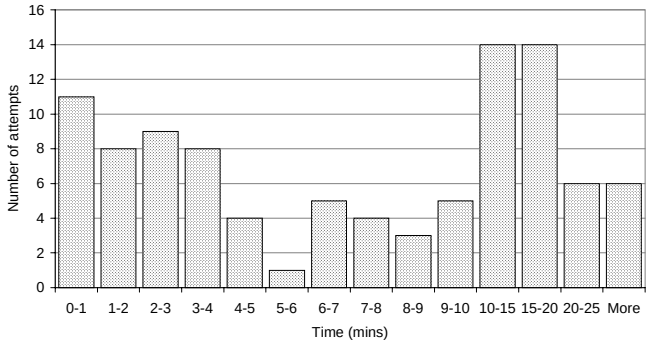


Figure 42: Time spent on attempts at conjecture C49

Case Study - Philip (Student S92)

Of all the students, Philip (S92) used Jape the most. Under 25 and with a BTEC in National Computing rather than A-levels, Philip already had a fair amount of programming experience before starting his Computer Science degree course. Learning to program was his main motivation for doing the course. His mathematics was weak (25% in the maths test).

Philip's score in Logic1 (54%) was below average. However, his scores in Logic2 (82%) and the Exam (81%) were among the highest in the cohort. At the start of term, he had expected the logic course to be fairly difficult and fairly interesting; his view was unchanged at the end of term. His view of the logic course was that it was only slightly worthwhile.

His interactions with Jape are represented in graphical form over the next few pages to give an indication of the sort of data available for each student.

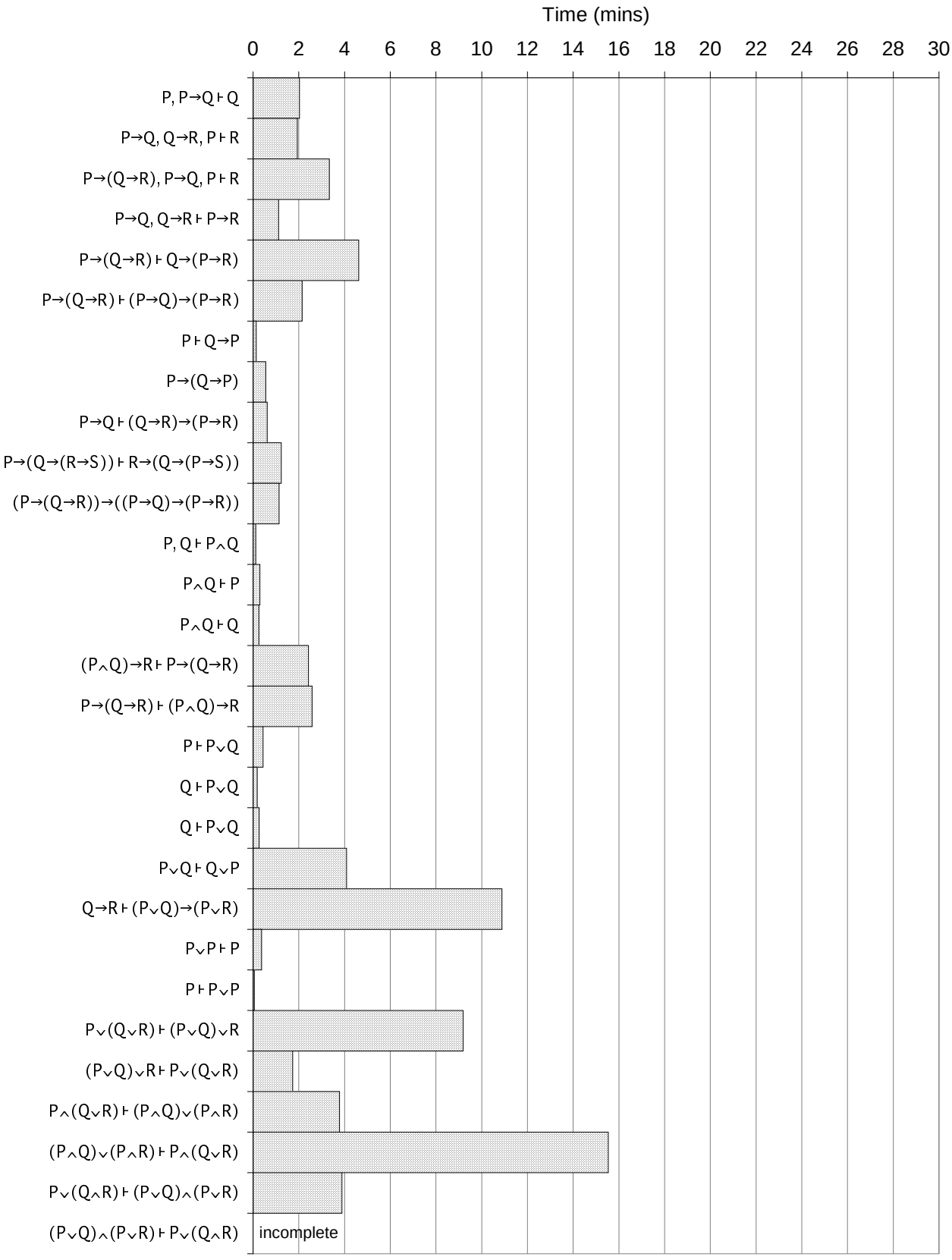


Figure 43: Student case study - Philip (S92) - 1st Jape session

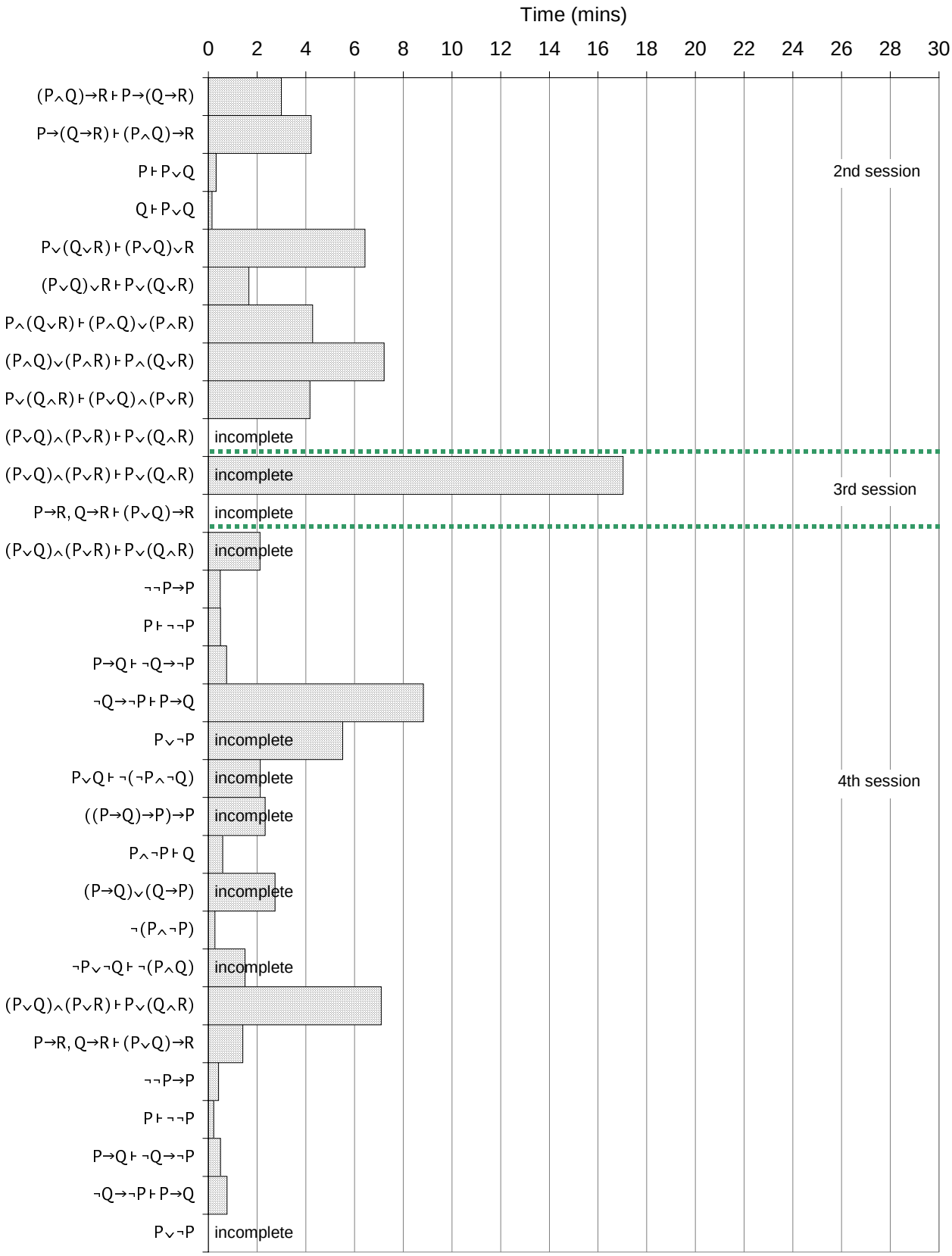


Figure 44: Student case study - Philip (S92) - 2nd, 3rd and 4th Jape sessions

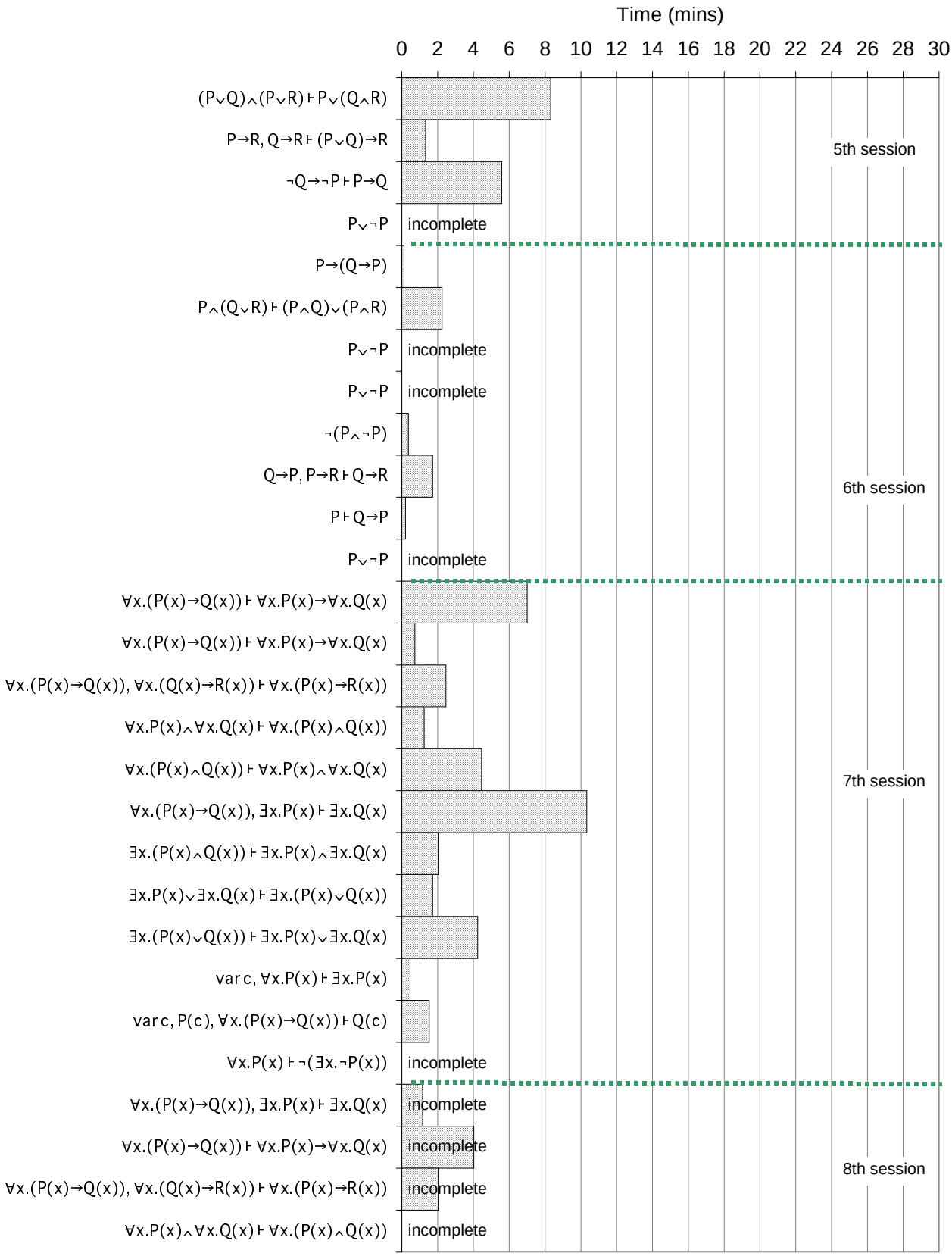


Figure 45: Student case study - Philip (S92) - 5th, 6th, 7th and 8th Jape sessions

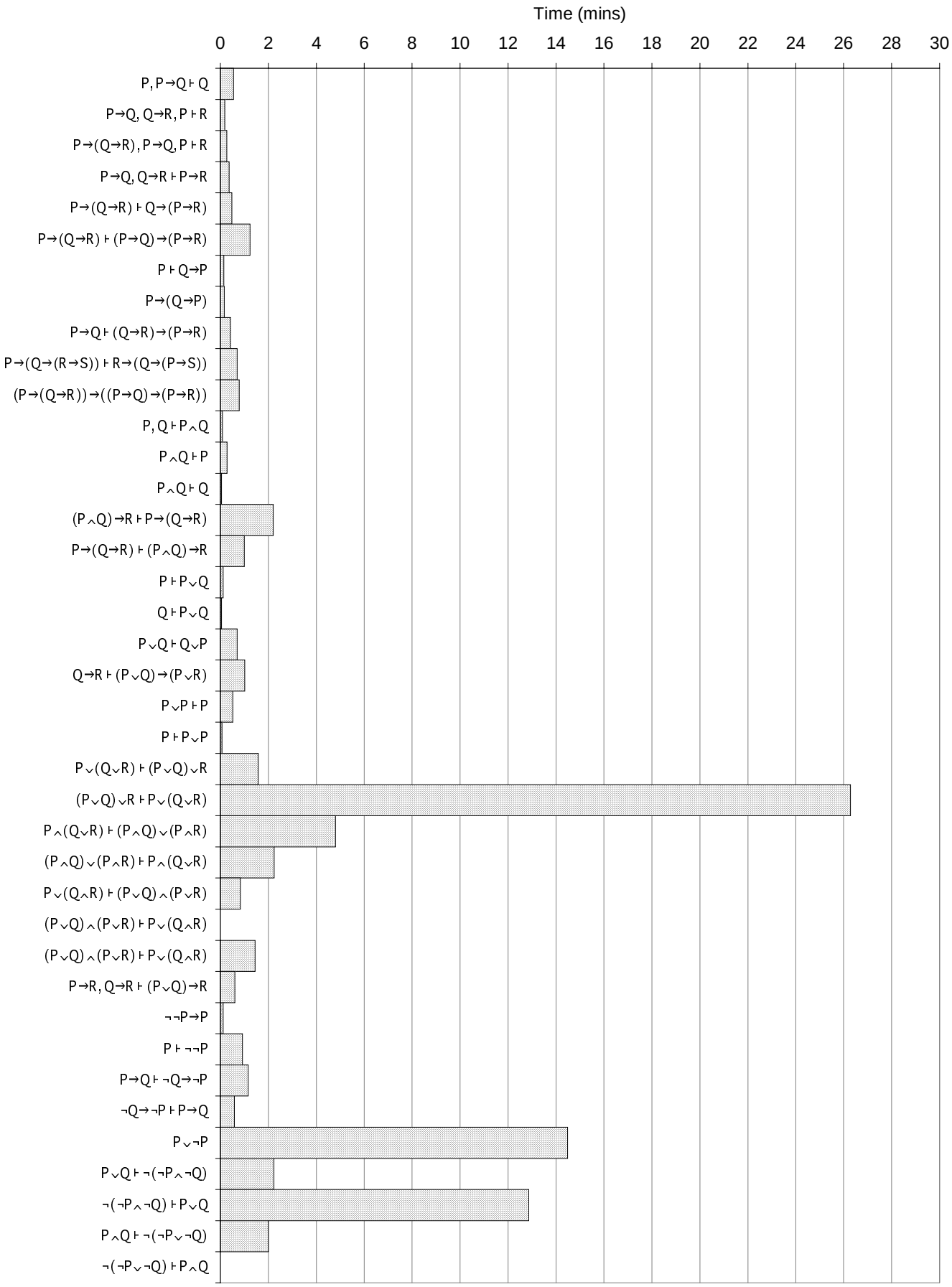


Figure 46: Student case study - Philip (S92) - 9th Jape session

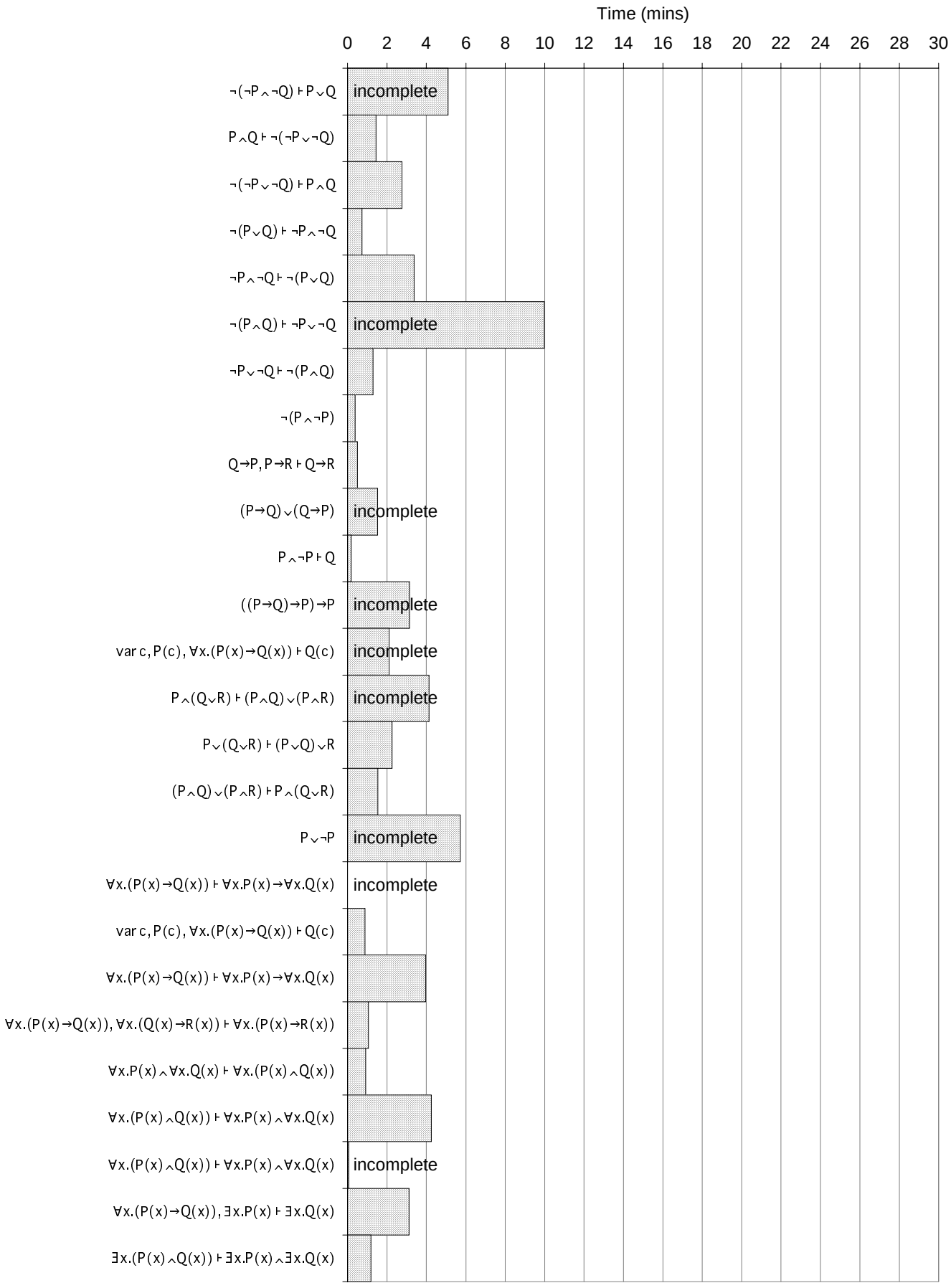


Figure 47: Student case study - Philip (S92) - 10th Jape session

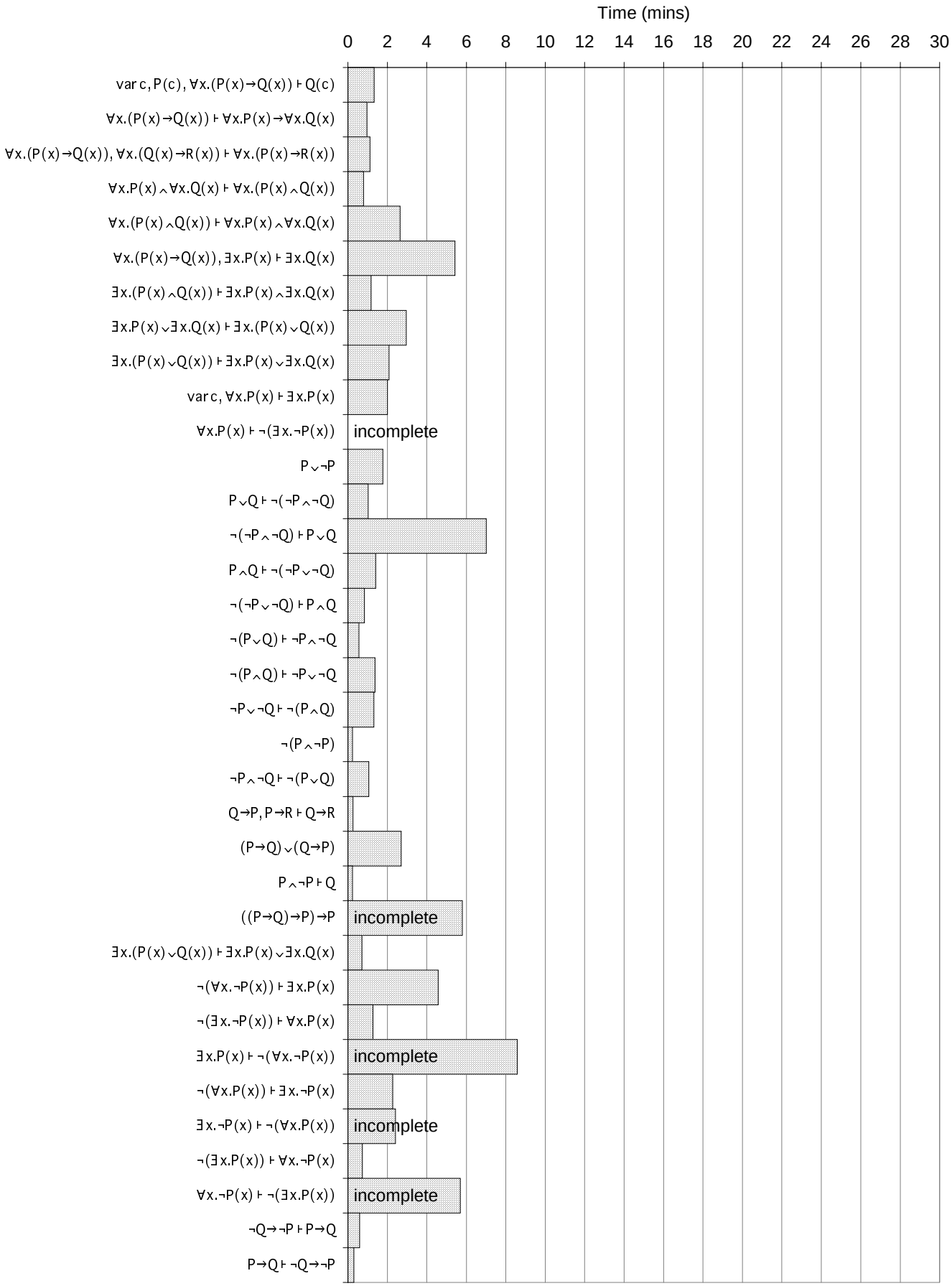


Figure 48: Student case study - Philip (S92) - 11th Jape session

Success Rate

Of all the proofs attempted, 70% were successfully completed; and the proportion of the total time spent on ultimately successful proof attempts was 67%. (Note however, there were 364 proof attempts that were abandoned, incomplete, but visible, at the end of a session. The times for these proofs are an underestimate; while the remaining 3951 proofs have accurate times. So, the actual proportion of the total time spent on successful proofs was rather lower than 67%.)

The closeness in value of the proportion of the proof attempts that were successfully completed and the proportion of the total time spent on ultimately successful proof attempts might suggest little difference in patterns of behaviour between successful and unsuccessful proofs. However, the following charts suggest that the situation is a little more complicated:

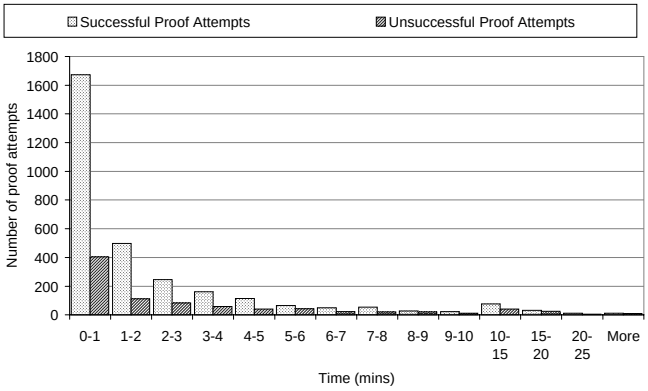


Figure 49: Histogram of proof attempts against time

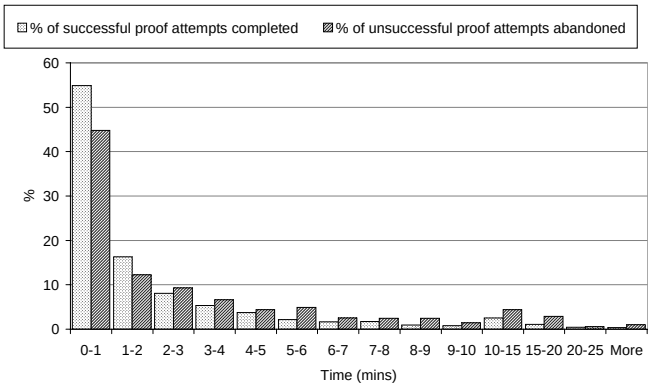


Figure 50: % of resolved proof attempts, by time

Moreover, the following chart suggests that unsuccessful proofs tend to take slightly longer on average than successful proofs:

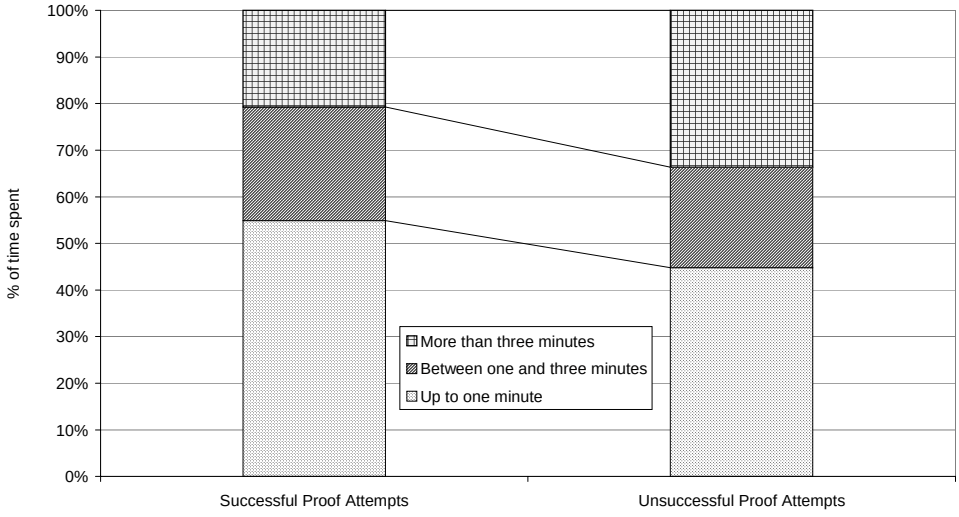


Figure 51: Duration of proofs

One might expect perhaps that if a student is faced with a difficult-looking conjecture, the proof attempt will be abandoned quickly. If it doesn't look difficult, the proof attempt will continue. One could then guess that the attempt will take less time if it is ultimately successful than if it is ultimately unsuccessful.

The following charts do not corroborate this hypothesis:

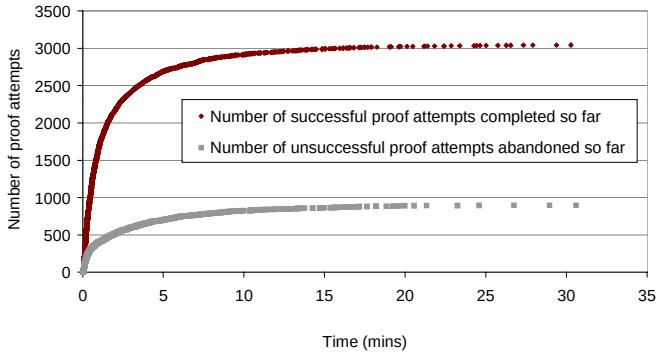


Figure 52: Cumulative chart of number of resolved proof attempts, by time

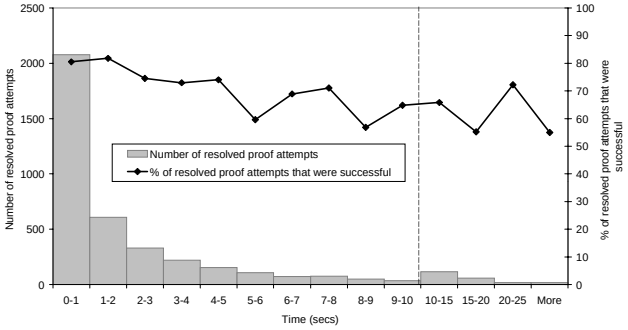


Figure 53: Success rate of proof attempts, up to 25 minutes

Nevertheless, the following chart suggests that of the proof attempts resolved in the first 5 seconds, about half are abandoned. The proportion of resolved proof attempts that are successful then leaps to almost 80% and peaks at about 90%. Then follows the slow downward trend that is continued in Figure 53.

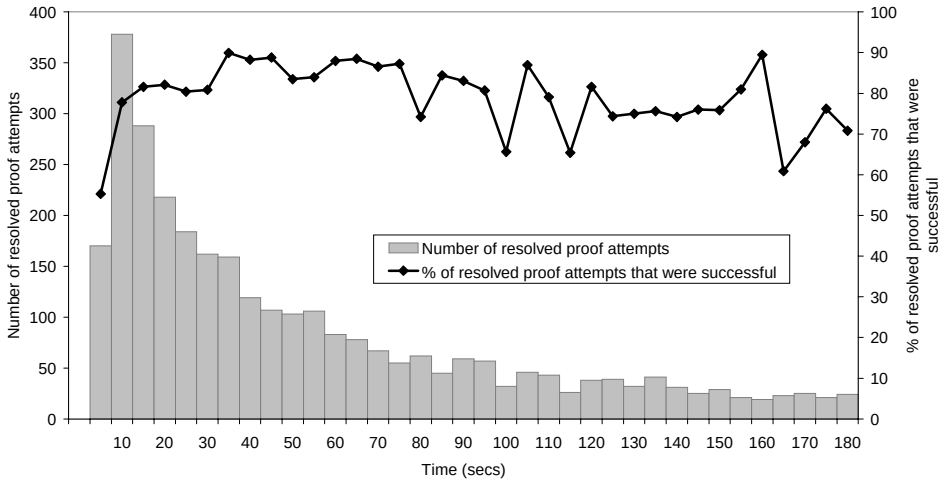


Figure 54: Success rate of proof attempts, up to 3 minutes

4.4.9 Jape usage - by topic

Number of students attempting each topic

There can be considered 7 topics - Implication, Conjunction, Disjunction, Negation, Quantifiers, False, and User-Entered Conjectures - however only the first five are of particular interest here. There are different numbers of conjectures in each topic.

Topic	Number of conjectures in the topic	Number of students attempting	Number of students who successfully proved at least one conjecture	Number of students who successfully proved all the conjectures	Number of students who proved all or all but one of the conjectures	Number of students who successfully proved at least 70% of the conjectures	Number of proof attempts	% of proof attempts that were successful	Number of proof attempts per student	Number of proof attempts per conjecture	% of proof attempts occurring in workshops	Time spent on proof attempts (mins)	% of time that is spent on successful proofs	Time per proof attempt (mins)	Time per successful proof (mins)	Time per unsuccessful proof (mins)	Number of proof attempts per hour	Number of successful proofs per hour
Implication	11	119	116	73	82	86	1601	89	13	146	76	2037	90	1.3	1.3	1.1	47	42
Conjunction	5	80	76	58	65	65	515	84	6	103	67	689	79	1.3	1.3	1.8	45	38
Disjunction	13	82	66	11	15	27	929	66	11	71	51	2770	54	3.0	2.4	4.0	20	13
Negation	18	53	41	0	2	3	552	40	10	31	30	1547	44	2.8	3.1	2.6	21	9
Quantifiers	18	83	64	0	0	2	680	52	8	38	75	2985	73	4.4	6.2	2.5	14	7
False	5	6					18				6	37						
User-entered	17	9					20				5	88						

Figure 55: Usage data by topic

Around 120 students attempted the first topic - Implication. Around 80 students attempted each of the second, third and fifth topics - Conjunction, Disjunction and Quantifiers. But only around 50 students attempted the fourth topic - Negation.

This pattern of student usage can be explained by reference to the three workshops. It was observed that most students in the first Jape lab session tackled all of Implication, and that many students completed Conjunction. The time available to students was determined by the length of the introduction to Jape at the start of the workshop, which varied for the four groups of students. There was also a network problem that made saving one's proofs problematic; but it was possible to give later groups instructions to circumvent the problem, saving them some of the unproductive time experienced by the first group. Some students had only 15 minutes remaining before other commitments started, others could stay for an hour. Some students, therefore, were also able to complete half of Disjunction. Most students did not have time to start Negation, however.

The second Jape workshop was optional, but it was observed that those that those students who attended started from where they left off in Conjunction or Disjunction, and proved four or five conjectures before making a start on Negation.

When the third workshop was run, students were advised to start Quantifiers, and most proved 3 or 4 conjectures. It was observed that there seemed to be much more assistance required in this workshop.

This pattern of activity is borne out by the data on proof attempts and times organised by date above: the vast majority of work on Implication, Conjunction and Disjunction occurred in the first workshop, the second workshop, between the first two workshops and in the period just before the Logic1 test. However, while there were some attempts on Negation at other times (in particular some limited activity during the second workshop),

the main activity on this topic is just before the Logic1 test - i.e. during students' own time. The main activity for Quantifiers occurs during the third workshop. Nearly all the work on False and User-Entered Conjectures was carried out just before the Logic2 test.

Almost 60% of students who used Jape successfully completed all the Implication conjectures. For Conjunction, this figure fell to 46%. Just 9% of students who used Jape (11 out of 125 students) finished Disjunction. This is somewhat surprising, given that almost two-thirds of students made a start on Disjunction. If one considers the students who successfully completed at least 70% of a topic, around 10% is added to these figures.

Nobody proved all of Negation. Nobody proved all of Quantifiers. Just 3 students proved more than 70% of Negation (Clement, S40; Joe, S67; and Philip, S92). Just 2 students proved more than 70% of Quantifiers (Philip, S92; and Keiko, S147).

Number of proof attempts at each topic

All the conjectures in each topic were attempted; however, the number of attempts per conjecture declined in each subsequent topic.

The following chart shows the success rate for proof attempts, by topic:

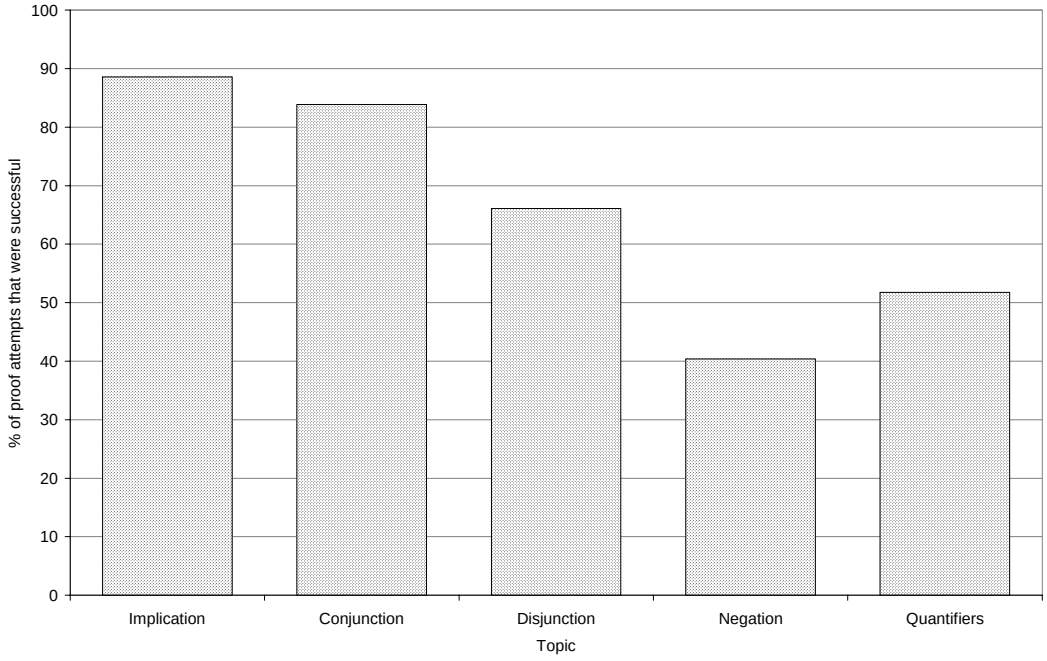


Figure 56: % of proof attempts that were successful, by topic

Judging by the proportion of *conjectures* that were completed successfully, Implication and Conjunction were the easiest topics, then Disjunction, then Quantifiers and Negation.

Time spent on each topic

The following chart shows the proportion of time spent on each topic.

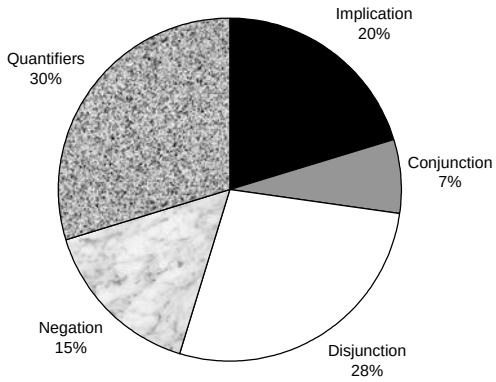


Figure 57: Proportion of time spent on each topic

The following chart shows the time spent per proof attempt, by topic. Taken with Figure 56, it suggests not just that Disjunction is harder than Implication and Conjunction, but that students are quicker to give up on Negation than Disjunction (one relevant factor here would be the proportion of time spent on each topic in workshops as opposed to students' own time, because human help would be available in workshops).

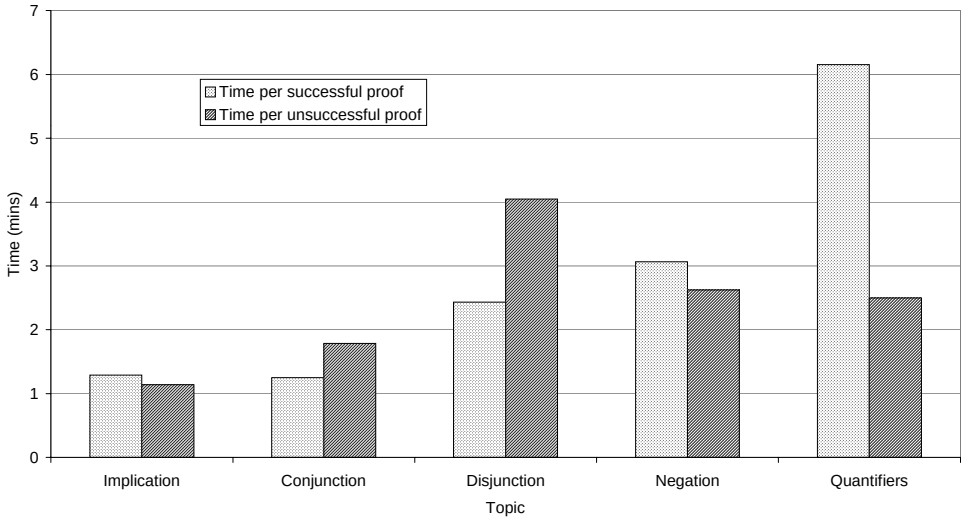


Figure 58: Time per proof attempt, by topic

Even though a greater proportion of Disjunction than Quantifiers was completed, proportionally *less* time is spent on successful proofs in Disjunction than in Quantifiers.

Students spent an average of over 6 minutes on successful Quantifier proof attempts, yet the mean time before abandoning a proof (2½ minutes) was much the same as for Negation.

Implication and Conjunction conjectures were successfully proved at a rate of about 40 an hour; Disjunction proofs at around 13 an hour; Negation proofs at 9 an hour; Quantifier proofs at 7 an hour.

4.4.10 Jape usage - by student

Number of proofs attempted by each student

This chart shows the full variation of students' commitment to working on Jape:

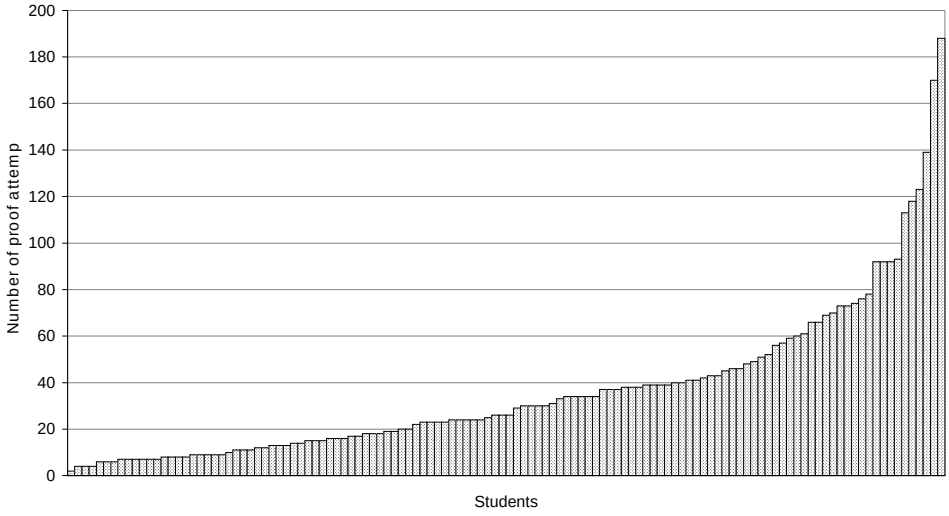


Figure 59: Number of proofs attempted by each student

The distribution of the number of *conjectures* attempted by each student is similar to the distribution of the number of proof attempts.

The students who attempted the largest number of proofs (above 80) are shown in the table below:

Student Number	Student Name	Number of proof attempts	% of proof attempts that were successful	Number of attempted conjectures (excl. false & user-entered)	% of attempted conjectures that were proved	% of all conjectures that were proved	% of Implication conjectures that were proved	% of Conjunction conjectures that were proved	% of Disjunction conjectures that were proved	% of Negation conjectures that were proved	% of Quantifier conjectures that were proved
92	Philip	188	80	65	92	92	100	100	100	94	78
60	Faye	170	75	44	89	60	100	100	100	39	17
147	Keiko	139	61	63	78	75	100	100	92	44	72
20	Marion	123	68	61	84	78	100	100	100	67	56
67	Joe	118	75	58	97	86	100	100	100	94	56
19	Charlotte	113	40	56	73	63	100	100	100	11	56
94	Marcel	93	57	46	76	54	100	100	54	22	44
40	Clement	92	59	63	86	83	100	100	92	78	67
107	Caroline	92	71	50	92	71	100	100	100	33	61
7	Nelson	92	45	37	89	51	100	100	54	0	56

Figure 60: Students who made the largest number of proof attempts

Eamon (Student S163) had a difficult time. He used Jape during the first workshop for just 10 minutes - proving the first four conjectures. He then used Jape during the third workshop, starting from conjecture C49 - ∀x.(P(x)→Q(x)) ⊢ ∀x.P(x)→∀x.Q(x). Here he spent 40 minutes attempting C49 on no less than 16 occasions (he was successful on the last), C50 on 9 occasions (without success), and C48 five times (successful twice). That was the sum total of his Jape experience.

The next highest number of attempts at the same conjecture was by Faye (S60) and Nelson (S7). Faye attempted C23 P∨(Q∨R)⊢(P∨Q)∨R on 10 occasions (3 successful), for a total of 34 minutes. Nelson attempted C58 ∀x.P(x) ⊢ ¬(∃x.¬P(x)) on 10 occasions (none successful) for at least half an hour (some times are not known).

Success rate

The success rate of proof attempts varied as follows:

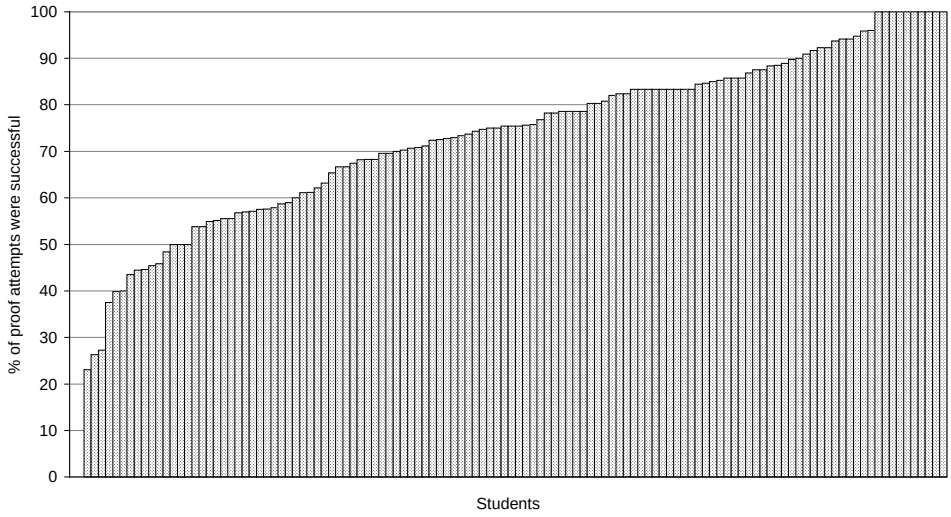


Figure 61: Students' success rate

Lou (student S116) had the worst success rate - 12 proof attempts without success. Wilfred (student S145) had 8 proof attempts without success. Amongst the high-usage students, Charlotte (student S19) and Nelson (student S7) had the lowest success rates, at 40% and 45% respectively.

Nearly all the students proved at least half the conjectures they attempted; and only around 15% of students left unproved more than a third of the conjectures they attempted.

The relationship between the number of proofs attempted by a student and the success rate is unclear:

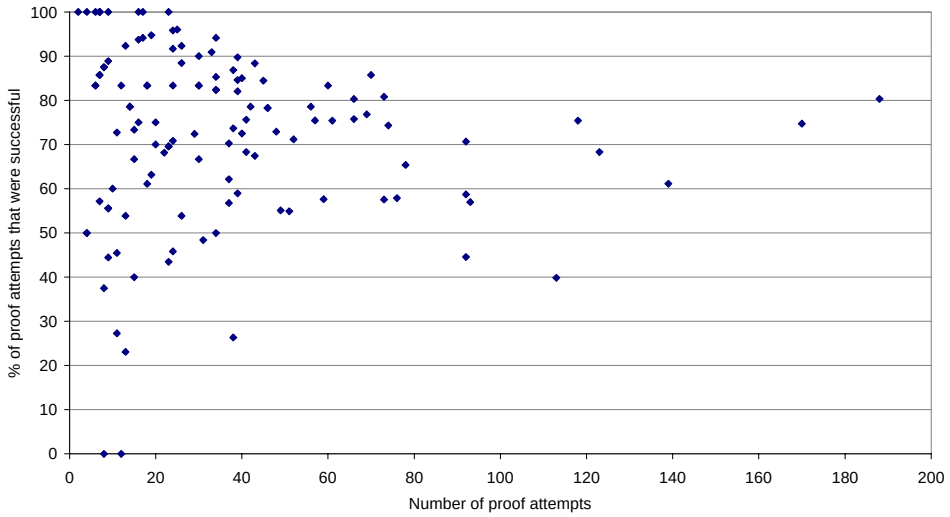


Figure 62: Relationship between the number of proof attempts and success rate

However, there is a high correlation (Pearson=0.96) between the number of proofs attempted and the number of successful proofs. Regression analysis suggests a fairly linear relationship $y=0.7x$. Moreover, the number of *proven conjectures* appears to depend strongly on the number of proof attempts:

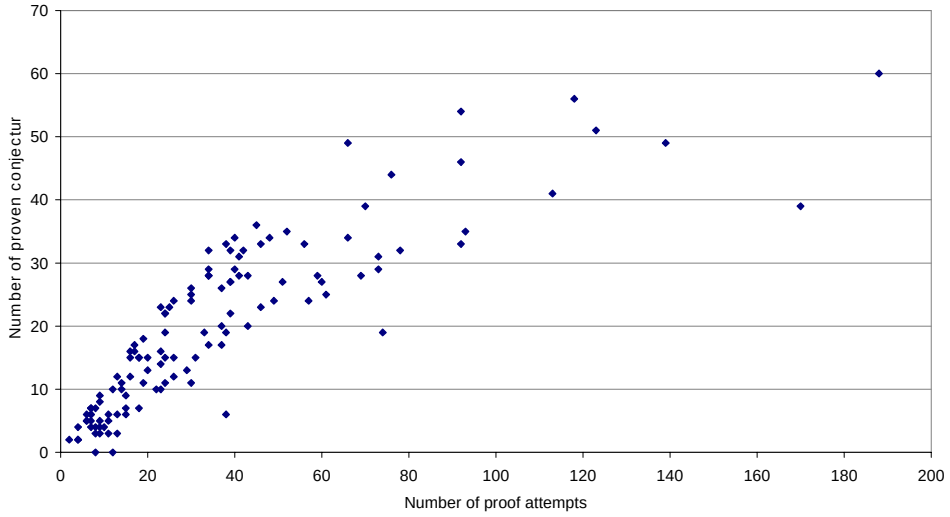


Figure 63: The relationship between the number of proof attempts and the number of proven conjectures

Although most students proved at least half the conjectures they attempted, the following chart shows how few students proved more than half the conjectures pre-installed in ItL Jape:

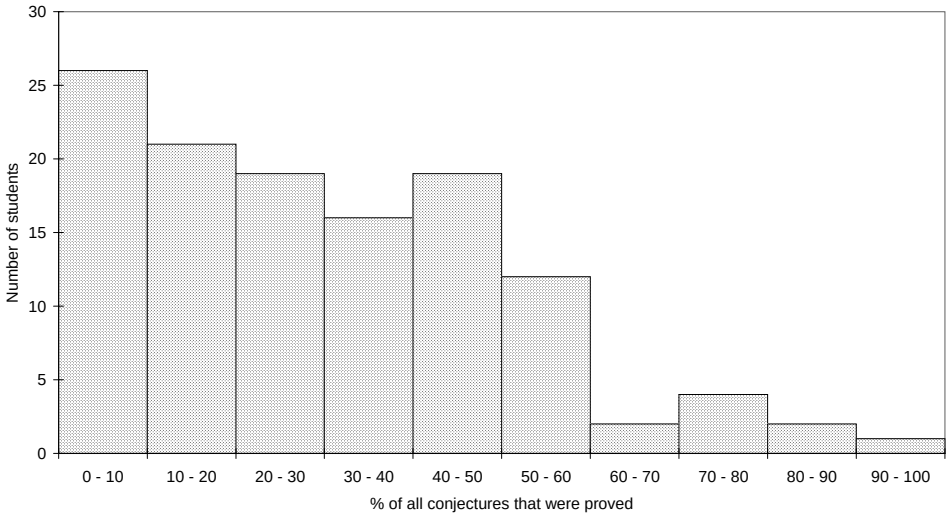


Figure 64: Histogram of the proportion of conjectures proved

The seven students who completed more than 70% of the conjectures in ItL Jape are listed in the following table:

Student Number	Student Name	Number of proof attempts	% of proof attempts that were successful	Number of proven conjectures	% of all conjectures that were proved	Time spent attempting proofs (mins)	Time per successful proof (mins)	Time per unsuccessful proof (mins)	Number of proof attempts per hour	Number of successful proofs per hour	Number of unsuccessful proofs per hour
92	Philip	188	80	60	92	443	2.2	2.9	25	20	5
67	Joe	118	75	56	86	245	2.2	1.6	29	22	7
40	Clement	92	59	54	83	195	1.7	2.7	28	17	12
107	Caroline	92	71	52	80	267	2.1	4.8	21	15	6
20	Marion	123	68	51	78	189	1.3	1.9	39	27	12
147	Keiko	139	61	49	75	276	1.9	2.1	30	18	12
62	Fred	66	76	49	75	169	1.9	4.5	23	18	6

Figure 65: Students who completed at least 70% of the conjectures given in ItL Jape

Time spent using Jape by each student

The chart for the time spent on Jape by each student is similar to that for the number of proofs, above. It is also possible to see the proportion of time which students spent working on proofs that were ultimately successful:

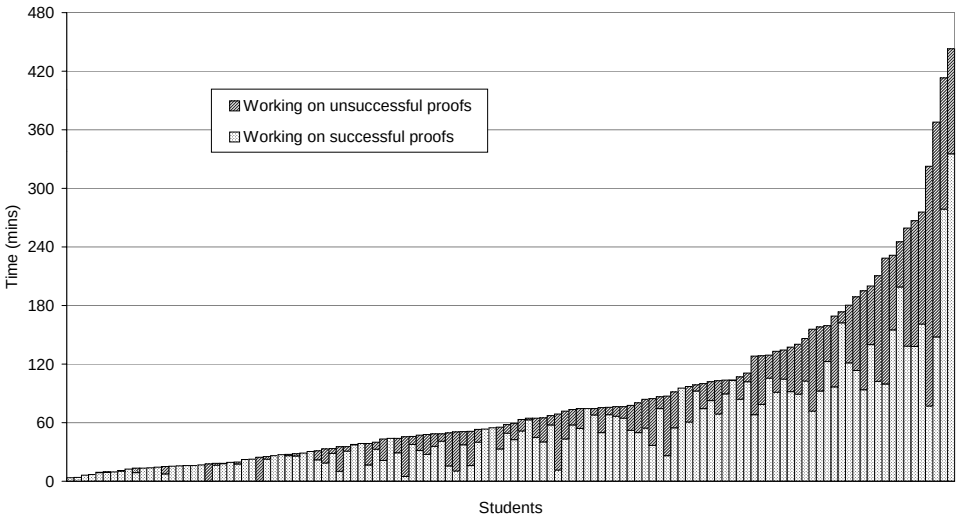


Figure 66: Time spent on Jape by each student

As the following chart shows, fewer than half the students spent over an hour using Jape, and only around 15% spent over 2½ hours.

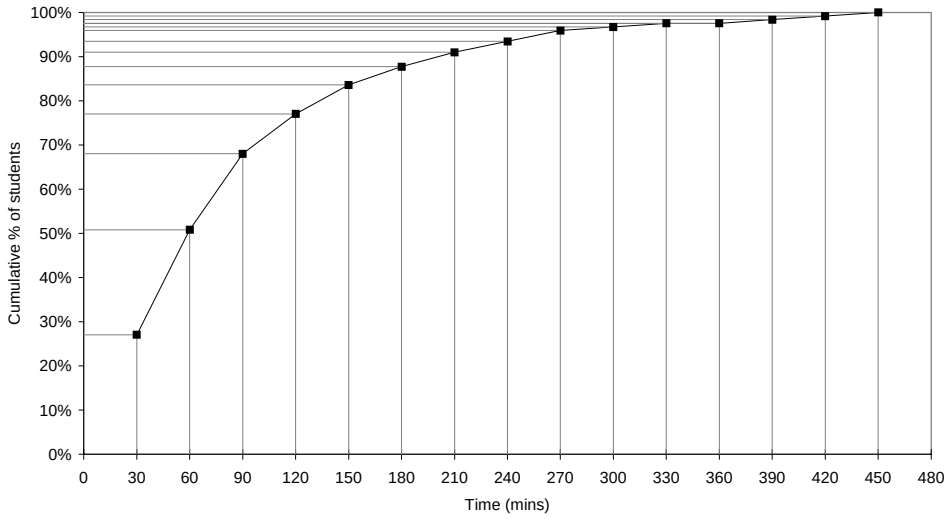


Figure 67: Time spent using Jape - cumulative %

The students who used Jape for the most time are shown in the table below:

Student Number	Student Name	Time spent attempting proofs	Number of days Jape was used	% of time that is spent on successful proofs	Number of proof attempts	% of all conjectures that were proved	Average time spent on each attempted proof (mins)	Average time spent on each successful proof (mins)	Average time spent on each unsuccessful proof (mins)	Average number of successful proofs per hour	Average number of unsuccessful proofs per hour	Average number of proven conjectures per hour
92	Philip	443	11	76	188	92	2.4	2.2	2.9	20	5	14
60	Faye	413	12	67	170	60	2.4	2.2	3.1	18	6	9
7	Nelson	368	10	40	92	51	4.0	3.6	4.3	7	8	9
22	Mat	323	6	24	37	26	8.7	3.3	17.6	4	3	5
147	Keiko	276	9	58	139	75	2.0	1.9	2.1	18	12	18
107	Caroline	267	6	52	92	71	2.9	2.1	4.8	15	6	17
134	Reg	260	6	53	78	49	3.3	2.7	4.5	12	6	12
67	Joe	245	8	81	118	86	2.1	2.2	1.6	22	7	23
38	Malcolm	231	7	67	76	68	3.0	3.5	2.4	11	8	19
89	Donald	229	6	44	41	43	5.6	3.6	9.9	7	3	12
19	Charlotte	210	10	49	113	63	1.9	2.3	1.6	13	19	19

Figure 68: Students who used Jape the most time (more than 3½ hours)

Levels of usage & progress

In the absence of natural cut-off points, **levels of usage** depending on time have been defined as follows:

Level of usage	Time spent using Jape	Number of students	Number of proof attempts (range)	% of all conjectures proven (range)
Zero	did not use Jape	24	0	0
Low	up to 1 hour	62	2 to 41	0 to 51
Medium	1-2 hours	32	9 to 69	5 to 55
High	more than 2 hours	28	37 to 188	26 to 92

Figure 69: Levels of Jape usage

Levels of progress depending on the proportion of all the ItL Jape conjectures proved have been defined as follows:

Level of progress	Proportion of all conjectures proved	Number of students	Number of proof attempts (range)	Time spent using Jape (range)
Zero	0	26	0 [1]	0 [2]
Low	less than a third	66	2 to 43 [3]	3 mins to 5½ hours
Medium	between a third and two-thirds	46	23 to 170	25 mins to 7 hours
High	more than two-thirds	8	66 to 188	2½ to 7½ hours

Figure 70: Levels of Jape progress

The following chart shows the variation in the relationship between progress and usage:

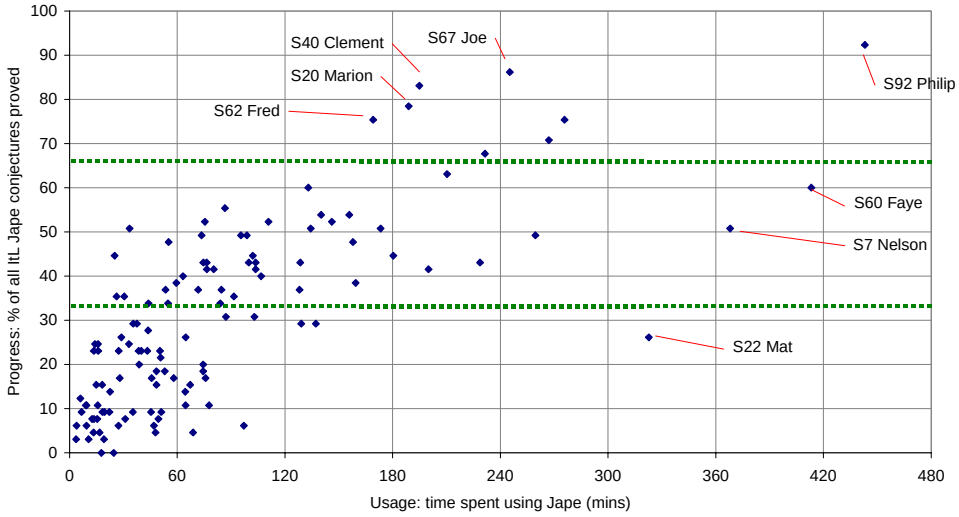


Figure 71: The relationship between usage and progress for students who used Jape

The students who spent a lot of time attempting proofs but failed to make much progress could be said to have found Jape difficult - they include Mat (S22, usage = 5½ hours, progress = 26%), Lucy (S21, usage = 2¼ hours, progress = 29%) and Christian (S174, usage = 2 hours, progress = 29%).

The students who made a great deal of progress mostly used Jape for more than three hours - they include Philip (S92, progress = 92%), Joe (S67, progress = 86%), Clement (S40, progress = 83%), Marion (S20, progress = 78%), Keiko (S147, progress = 75%), Fred (S62, progress = 75%), Caroline (S107, progress = 71%), and Malcolm (S38, progress = 68%).

Proof speed

Although the correlation between progress and usage is 0.73, spending a long time on Jape is clearly no guarantee of success. Of the students who spent most time using Jape, Faye, Nelson and Mat are not in the list of students who proved more than 70% of the conjectures in ItL Jape.

Conversely, it is possible to prove a large proportion of the conjectures without spending much more than 3 hours using Jape - for example, Clement, Marion and Fred.

---

[1] Excluding Wilfred (S145) and Lou (S116) - discussed earlier - who between them attempted 20 proofs without success.

[2] Excluding Wilfred (S145) and Lou (S116), who between them spent over 40 minutes using Jape without proving a single conjecture.

[3] In fact one student - Lucy (S21) attempted 74 proofs, but proved only 19 conjectures. This is an outlier.

There is a slightly higher correlation (0.88) between the number of proofs attempted and the time spent using Jape; however, a large number of proof attempts can be symptomatic of a *lack* of progress. Using the proportion of all conjectures proved as a measure of progress is more reliable.

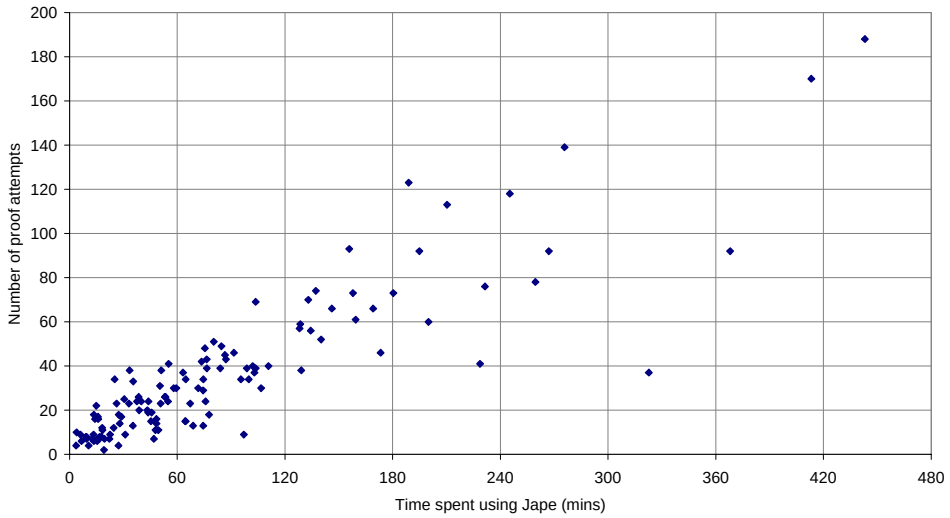


Figure 72: Relationship between the number of proof attempts and the time spent using Jape for all students who used Jape

The students who proved most conjectures tended to have average proof speeds, and average conjecture speeds. Of the students who made more than 20 proof attempts, the student who got through the most proofs an hour was Ben (S29) at 88 proof an hour. However, on average, 28 of these would be abandoned. The student with the fastest success rate was Gabriel (S161) at 69 successful proofs an hour. Gabriel also proved almost half the conjectures in ItL Jape; whereas Ben only proved 15%.

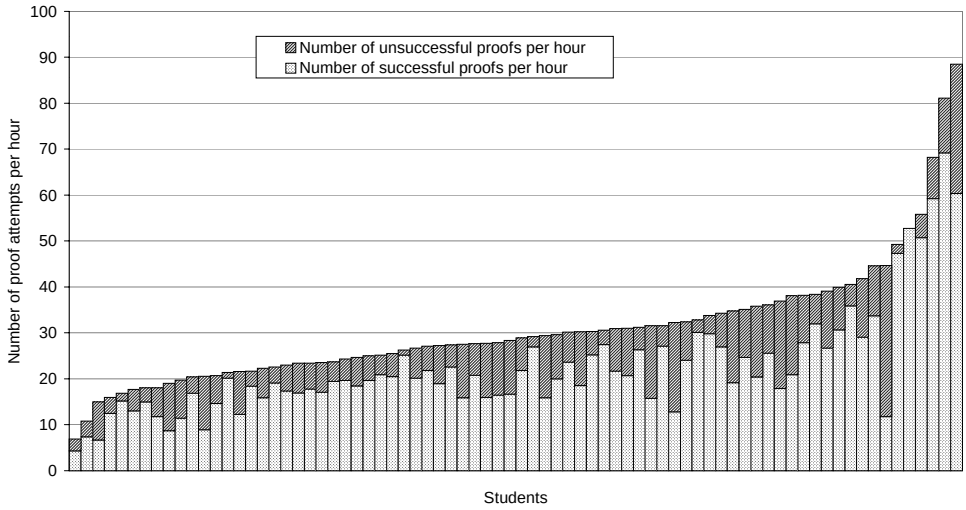


Figure 73: Proof speed, for students who made more than 20 proof attempts

The two students making the highest progress were Philip & Joe (S92 and S67 respectively).

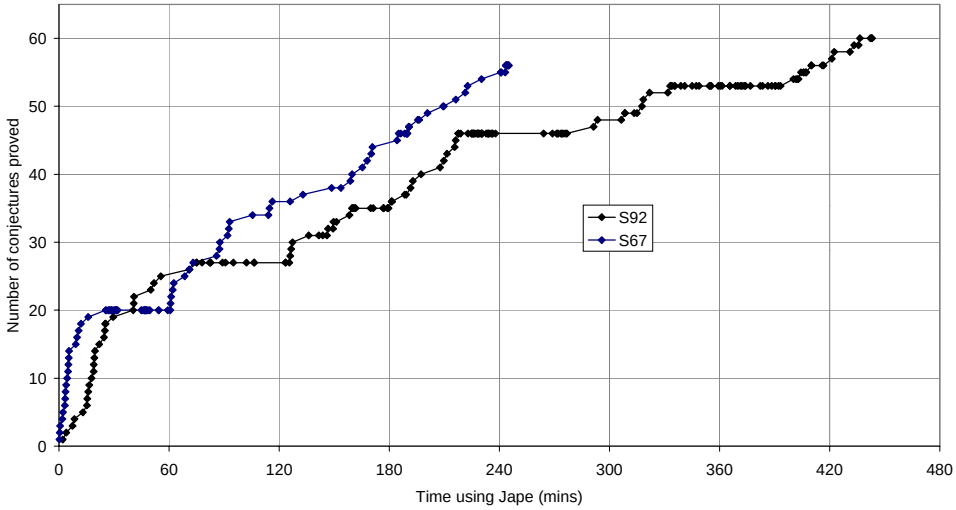


Figure 74: Number of conjectures proved over time for the two students making the most progress (S92, S67)

Another way to look at the data is by the time spent on each conjecture rather than by the number of conjectures proved by a certain time:

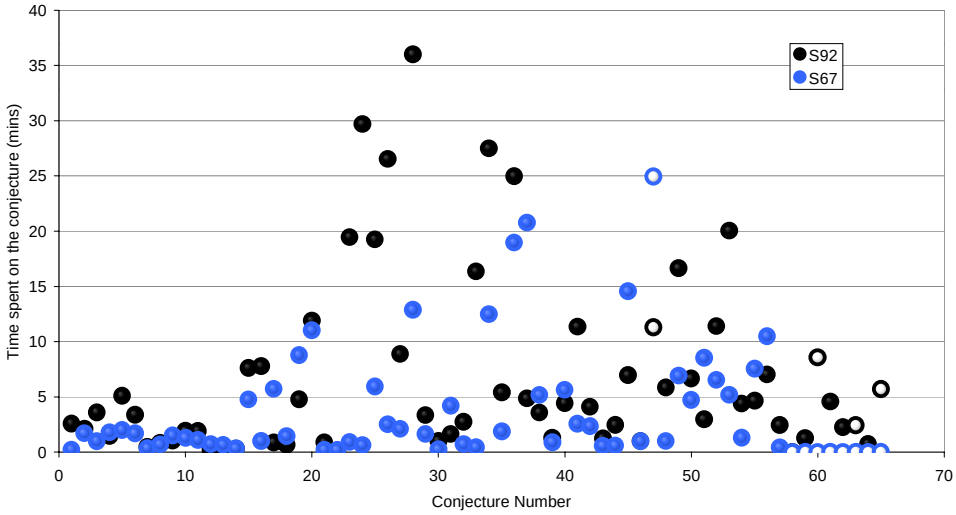


Figure 75: Comparison of the two students making the most progress in terms of the time spent on each conjecture (unfilled data points refer to unproved conjectures)

Case Study - Conjecture C34 - P ∨¬P

Conjecture C34 - P∨¬P - has the highest number of unsuccessful attempts of the conjectures in the Negation topic.

The following chart shows the time taken to first prove C34 by all students who attempted the conjecture. If a particular student did not prove the conjecture, the total time is shown.

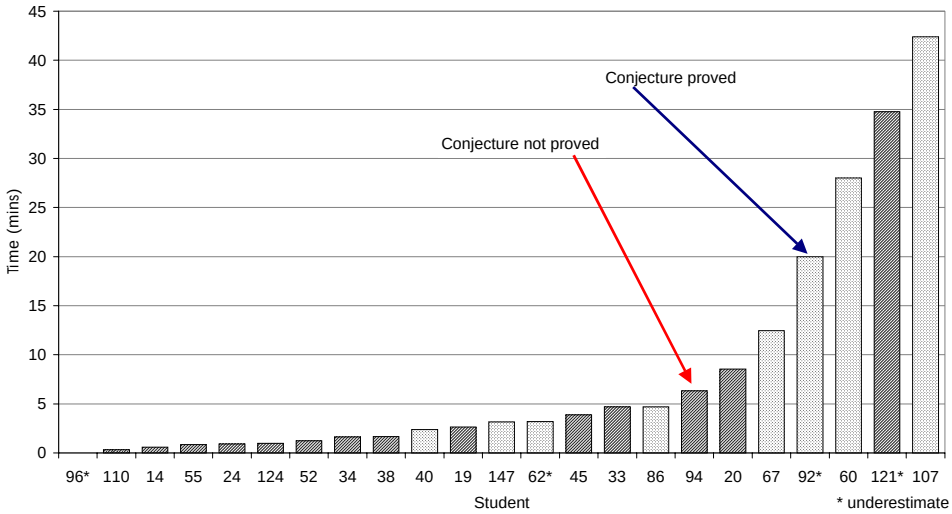


Figure 76: Time taken to first prove conjecture 34 - P∨¬P

Faye, Philip and Caroline (S60, S92 and S107 respectively) each proved the conjecture P∨¬P on two separate occasions. Faye took 28 minutes to prove it during the second Jape workshop. 9 months later (she was absent from Logic2 and the exam) it took her 17 minutes. Philip attempted the conjecture 9 times. For technical reasons, it is not known how long he spent on the conjecture on five of these occasions, but it *is* known that he did not prove it until the seventh attempt (taking 14½ minutes). Four days later, he attempted it again without success (taking 5½ minutes). Three days after that he proved it successfully, taking just under 2 minutes. Caroline spent almost half an hour in two unsuccessful attempts at the conjecture during the second Jape workshop. She succeeded on the third attempt, taking 15 minutes. Two days later she successfully proved it in 5 minutes.

The shortest time in which the conjecture P∨¬P was proved was 72 seconds (student S40, Clement). His only other attempt at this conjecture had been two days earlier, when he spent exactly the same amount of time unsuccessfully trying to prove it.

Joe, Toby and Keiko (S67, S86 and S147 respectively) each proved the conjecture in one attempt.

Fred (S62) proved the conjecture on his second attempt, taking 3½ minutes. However, he then failed to prove it on 7 subsequent occasions. This raises the possibility that he obtained help from a friend or from notes when he proved it, or that he found the proof by chance and could not reproduce the steps.

Progress through the topics

It is possible to explore in which topics progress tends to be manifested. For example, for those students whose level of progress was high, 22% of the conjectures they proved were within the Quantifiers topic. Just 12% of the conjectures proved by medium-progress students were within Quantifiers; and just 9% of the conjectures proved by low-progress students were within Quantifiers.

The contrast for Negation is even starker - 23% of the conjectures proved by high-progress students were within the Negation topic. The proportions for students making less progress are under 5%.

So in which topic are low-progress and medium-progress students making their progress? For the low-progress students, 71% of the conjectures proved were in the Implication topic. The comparable figure for the medium-progress students is 37% - they also prove 29% of their conjectures in Disjunction.

So it looks like the low-progress students do not get much beyond Implication and Conjunction; while the medium-progress students do not get much beyond Disjunction. Analysis of the times spent on the different topics shows that while the students belonging to each level of progress spent around a third of their time on Quantifiers

(because of the tutors' instructions at the beginning of the third Jape workshop to start with these conjectures), the low-progress students spent most of the rest of their time on Implication, the medium-progress students spent most of the rest of their time on Disjunction, and the high-progress students spent most of the rest of their time on Disjunction and Negation.

Number of days Jape was used

The high-progress students used Jape on an average of 7.6 days. The medium-progress students used Jape on an average of 4.1 days. The low-progress students used Jape on an average of 2.3 days.

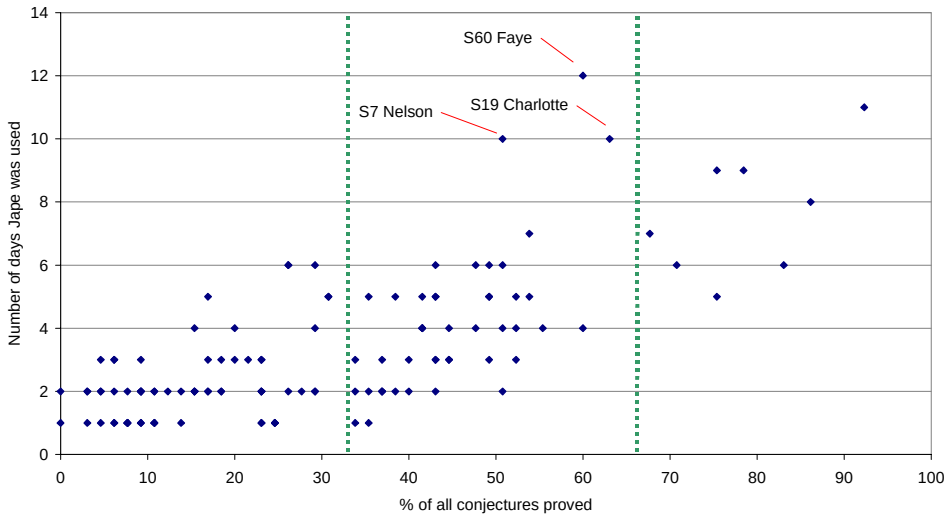


Figure 77: Relationship between proportion of conjectures proved and the number of days Jape was used

4.4.11 Jape usage - by student group

Nine categories of student groups are discussed here: gender, age, degree course, overseas status, entry qualifications, prior computer experience, motivations, course expectations, and research participation.

Gender

On average, the 25 females used Jape more than the 121males (Mann-Whitney U test: $p<0.001$ for time spent), even though this is not reflected in the estimates of their own usage that they gave in Survey2. A larger proportion of females than males were high-usage students (χ^2 test: $p=0.01$). 66% of males were either zero-usage or low-usage students, compared to only 24% of females. All females used Jape.

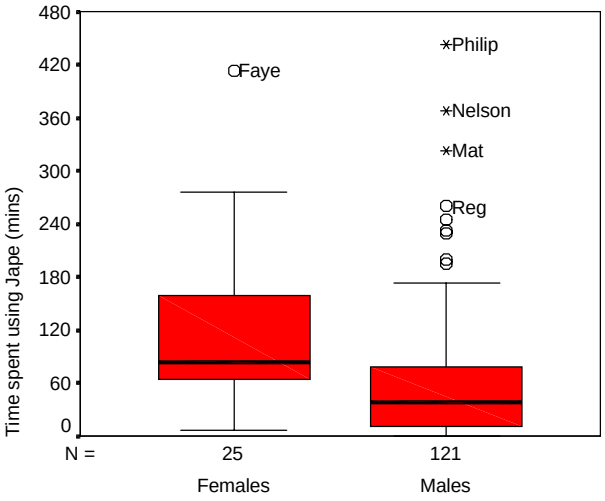


Figure 78: Time spent on Jape, for males and females

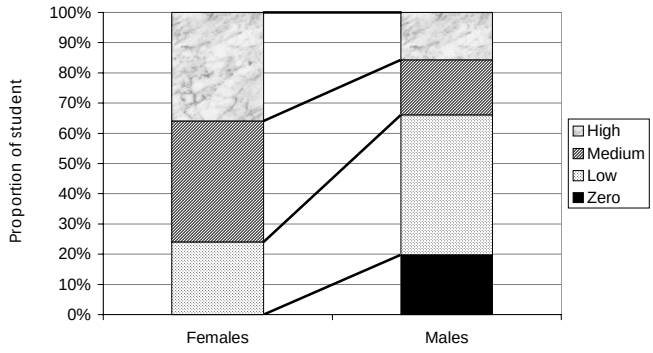


Figure 79: Levels of Jape usage, for males and females

On average, females made slightly more progress than males (Mann-Whitney U test: $p=0.004$, where progress is measured by the number of conjectures proved), however the advantage is not statistically significant at the 5% level when zero-usage students are removed. Nevertheless, females did attempt more of the Negation conjectures than males (Mann-Whitney U test: $p=0.022$), and proved more of the Quantifier conjectures ($p=0.026$).

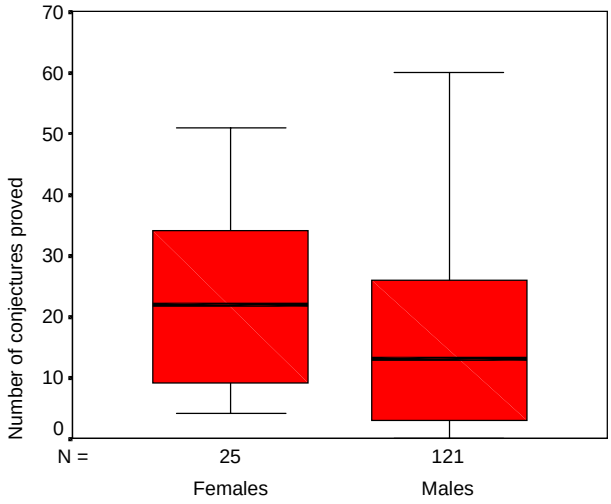


Figure 80: Number of conjectures proved using Jape, for males and females

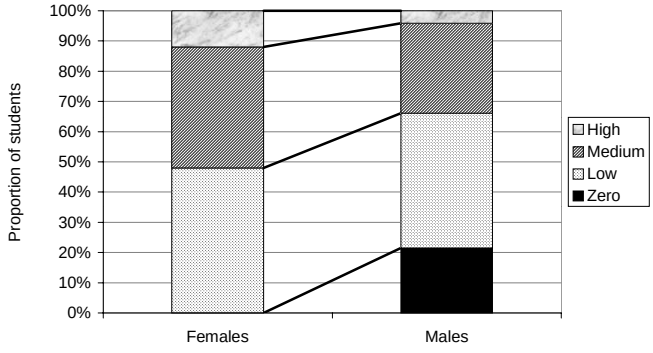


Figure 81: Levels of Jape progress, for males and females

Age

On average, the 11 older students used Jape more than the 132 younger students (Mann-Whitney U test: $p<0.001$ for time spent). Almost two-thirds of younger students either failed to use Jape or were low-usage students. No older students failed to use Jape, and just one (out of 11) was a low-usage student.

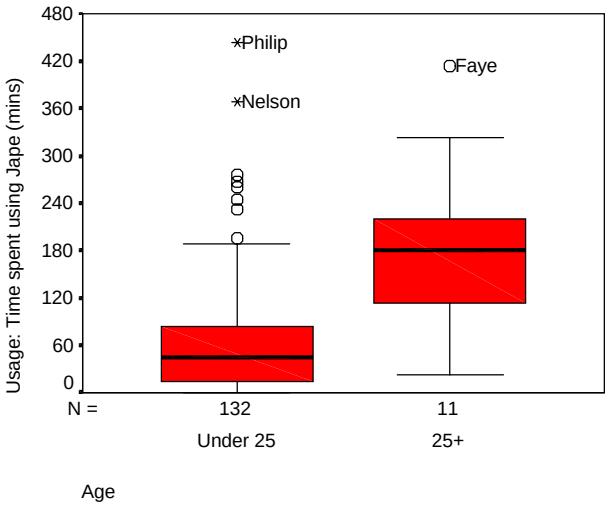


Figure 82: Time spent using Jape, by age

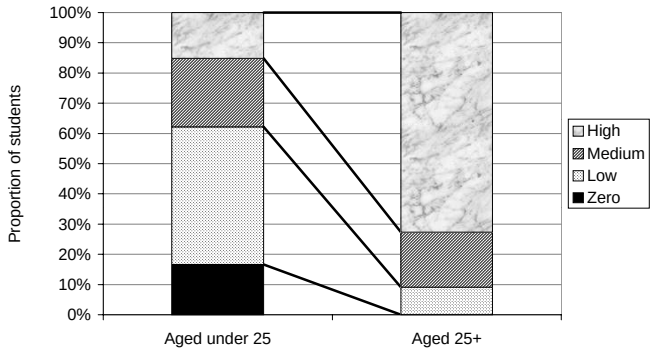


Figure 83: Levels of Jape usage, by age

However, there was no statistically discernible difference in progress between older and younger students (the median number of conjectures proved was 27 for the older students, and 14 for the younger students, but the sample size is small). Nevertheless, the older students attempted more Disjunction and Negation conjectures (Mann-Whitney U test: $p<0.05$). They spent more time on each proof on average (Mann-Whitney U test: $p=0.007$); but they spent a smaller proportion of their time on successful proofs ($p=0.008$); and they proved a smaller average number of successful proofs per hour ($p=0.008$).

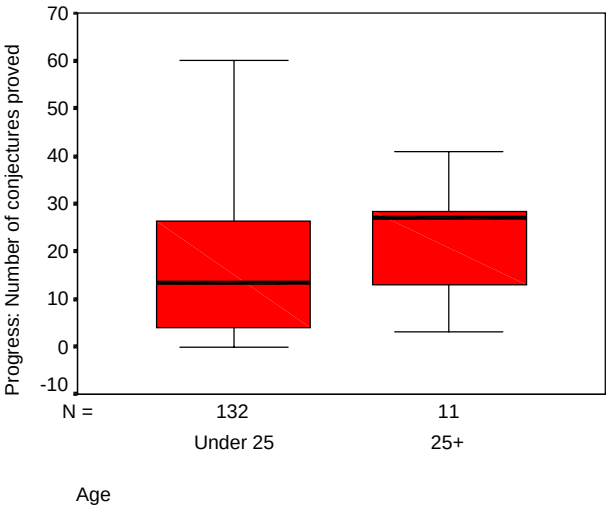


Figure 84: Number of conjectures proved using Jape, by age

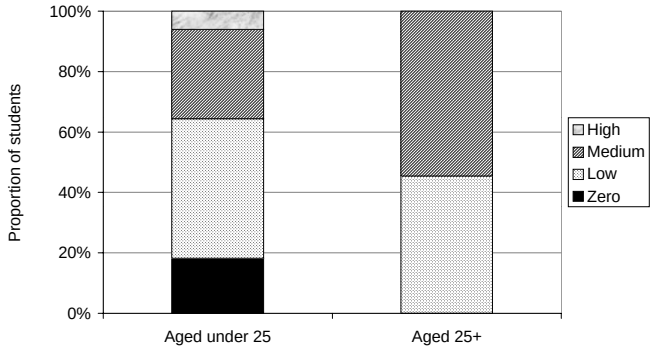


Figure 85: Levels of Jape progress, by age

Degree course

On average, the 104 students registered for the degree "Computer Science" used Jape more than the 42 students registered for other degrees (Mann-Whitney U test: p=0.022 for time spent). Only one student (out of 20) registered for "Computer Science and Business Studies" was a high-use student, compared to almost a quarter of "Computer Science" students, and just over a quarter of "Computer Science and Mathematics" students.

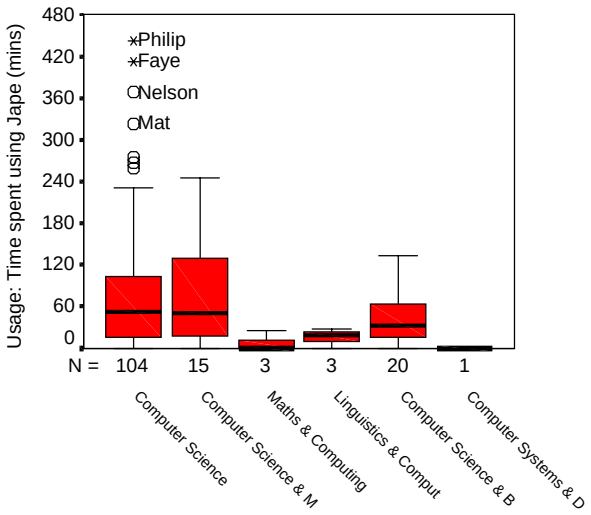


Figure 86: Time spent on Jape, by registered degree

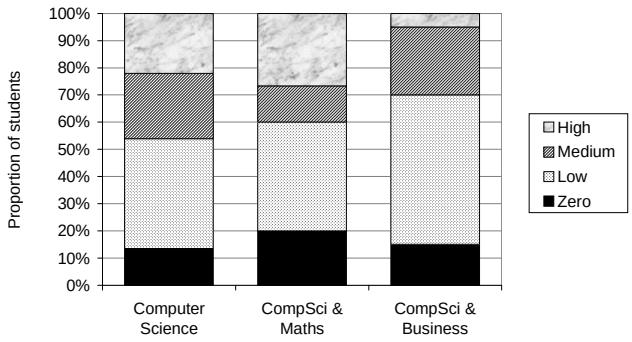


Figure 87: Levels of Jape usage, by registered degree

On average, students registered for the degree "Computer Science" made slightly more progress than the others (Mann-Whitney U test: p=0.024), however the advantage is not statistically significant at the 5% level when zero-usage students are removed. Students registered for "Computer Science and Business Studies" who used Jape tended to make less progress than other Jape users (p=0.015), particularly in Implication, Conjunction and Disjunction (p<0.05). However, they spent a much higher proportion of their time on successful proofs, which suggests that they did not attempt conjectures they thought difficult. For example, half the Computer Science students attempted more than 4 Disjunction conjectures; whereas almost 60% of the Computer Science & Business Studies students did not attempt Disjunction at all and none attempted more than four Disjunction conjectures.

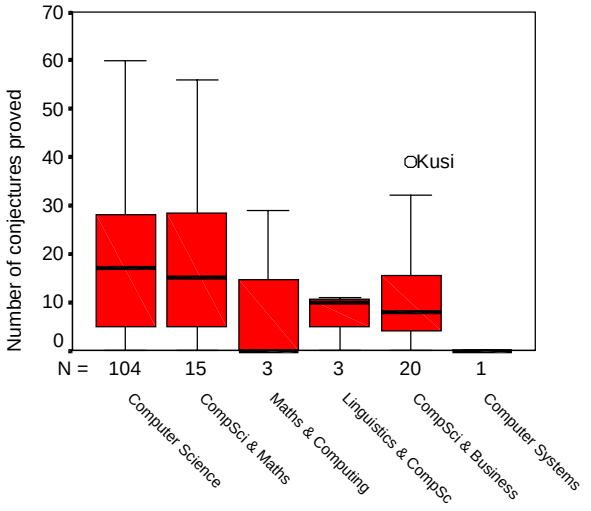


Figure 88: Number of conjectures proved using Jape, by registered degree

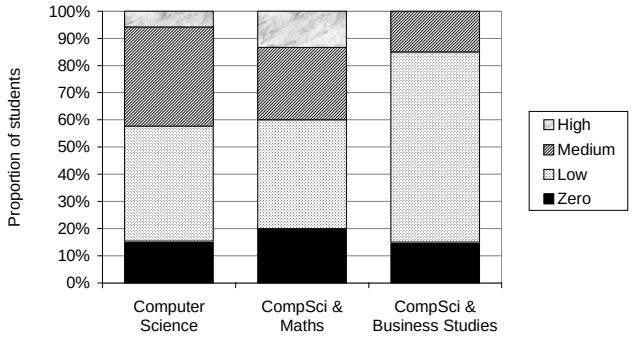


Figure 89: Levels of Jape progress, by registered degree

Overseas status

There seems to be little difference between overseas students' and other students in terms of Jape usage and progress.

Entry qualifications

On average, the 114 students with A-levels used Jape slightly less than the 29 students with other qualifications (Mann-Whitney U test: p=0.048 for time spent). The students with A-levels were less likely to be high-usage students (χ^2 test: p=0.003).

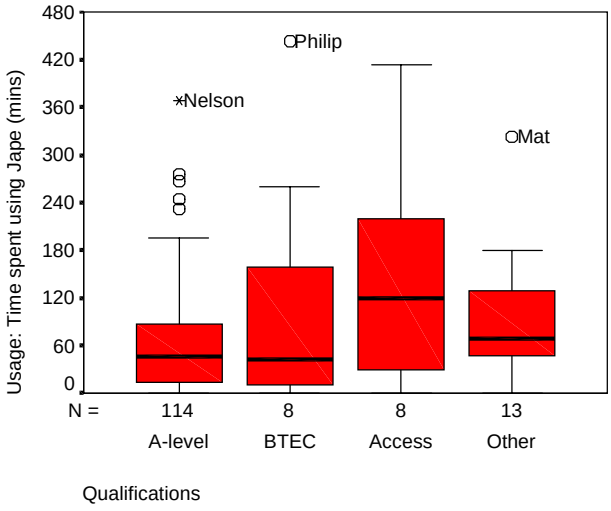


Figure 90: Time spent on Jape, by qualifications

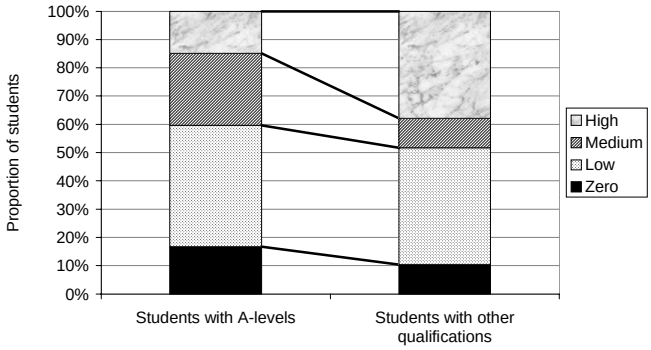


Figure 91: Levels of Jape usage, by qualifications

On average, there was no statistically discernible difference in progress between these two groups of students.

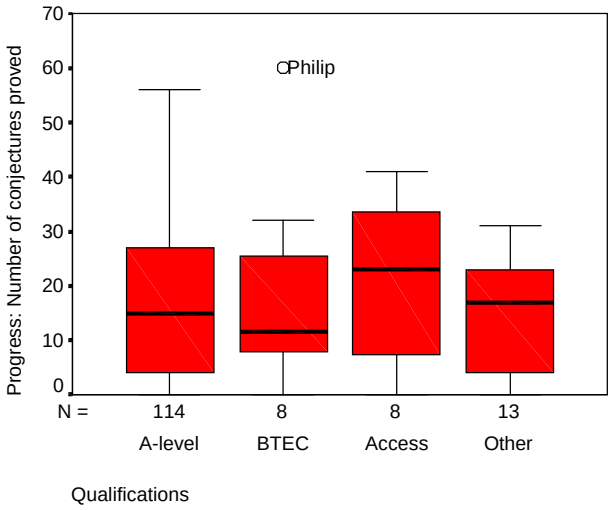


Figure 92: Number of conjectures proved using Jape, by qualifications

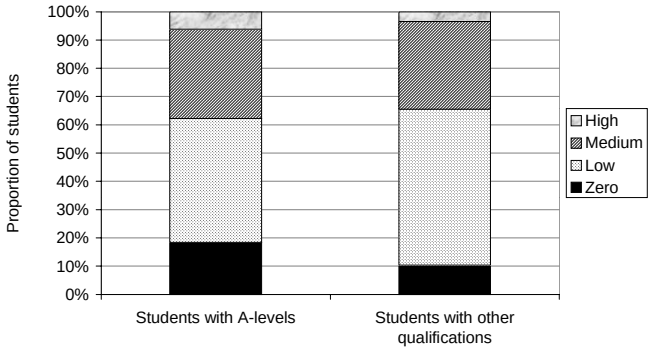


Figure 93: Levels of Jape progress, by qualifications

Prior computer experience

On average, the 52 students with no programming experience used Jape slightly less than the 72 others (Mann-Whitney U test: p=0.046).

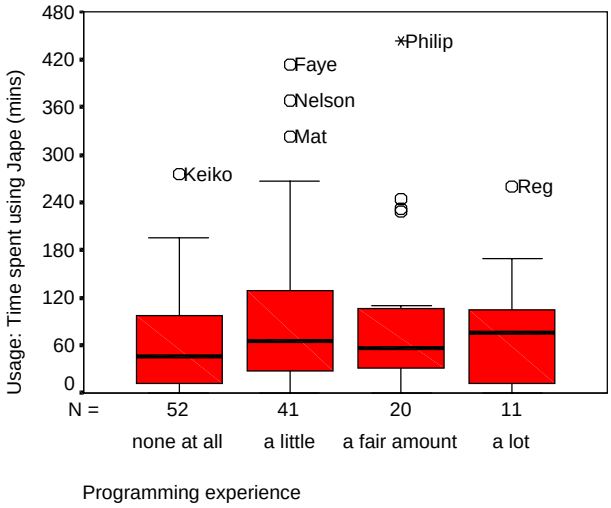


Figure 94: Time spent on Jape, by programming experience

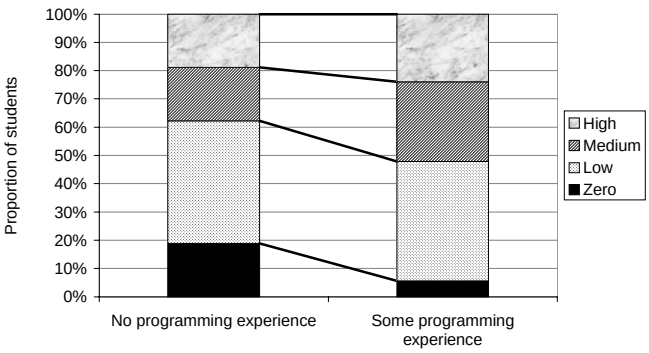


Figure 95: Levels of Jape usage, by programming experience

On average, the 52 students with no programming experience made slightly less progress than the 72 others (Mann-Whitney U test: p=0.044).

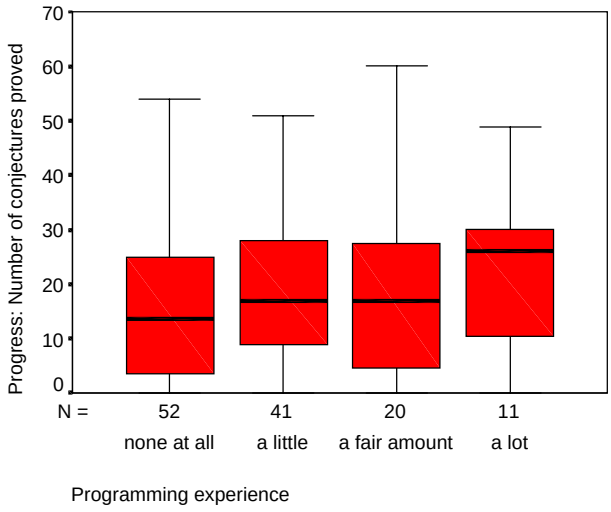


Figure 96: Number of conjectures proved using Jape, by programming experience

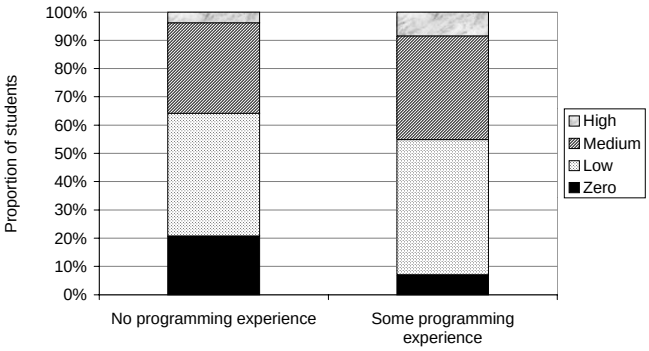


Figure 97: Levels of Jape progress, by programming experience

Students' other prior computer experiences do not appear to have much of an effect on Jape usage or progress.

Motivations

On average, the 17 students who indicated that their main motivation was "learning to program" used Jape slightly more than the 95 others (Mann-Whitney U test: p=0.038 for time spent) - the difference was mainly in the proportion of medium-usage students. Those who indicated "financial rewards" tended to spend less time.

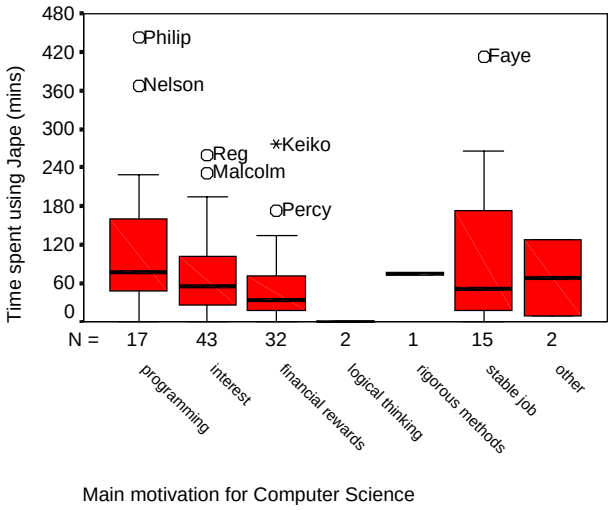


Figure 98: Time spent on Jape, by main motivation

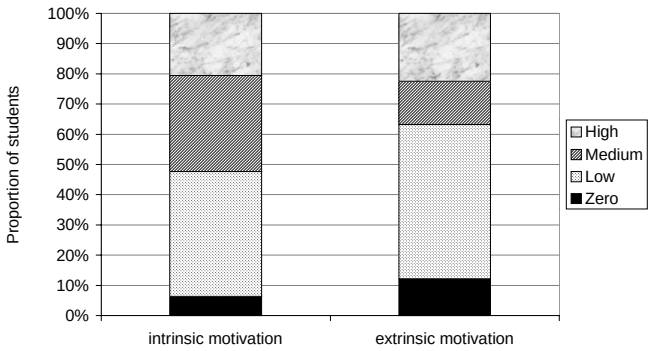


Figure 99: Levels of Jape usage, by main motivation

Considering the students who used Jape, on average the 16 students who indicated that their main motivation was "learning to program" made more progress in Quantifiers than the 86 others (Mann-Whitney U test: p=0.007).

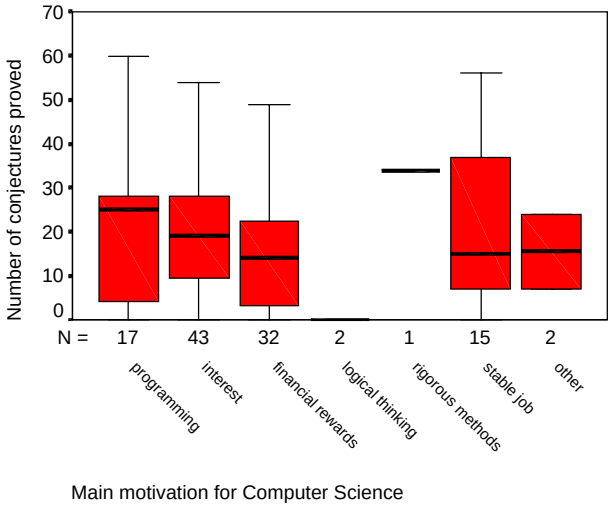


Figure 100: Number of conjectures proved using Jape, by main motivation

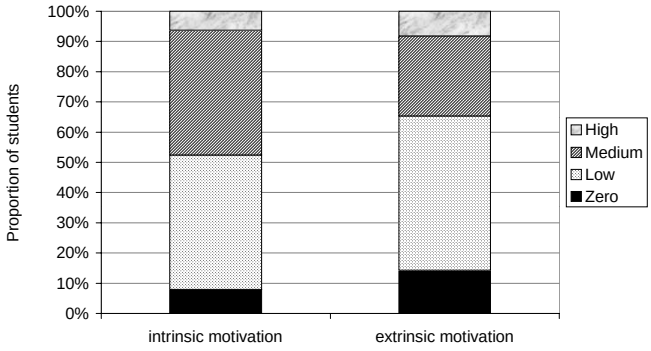


Figure 101: Levels of Jape progress, by main motivation

Course expectations

On average, the 9 students who expected the course to be "uninteresting" or "fairly uninteresting" used Jape much less than the 116 others (Mann-Whitney U test: p=0.002). Four of them did not use Jape at all, and the other five spent no longer than 75 minutes on Jape in total.

Students' expectations of the difficulty of the course do not appear to have much of an effect on Jape usage or progress.

Research participation

On average, the 11 participants in the research (i.e. those who volunteered to take part in either the Observational Study or the Reflection Study) used Jape more than the 111 students who those who used Jape but didn't take part

in the research (Mann-Whitney U test: $p<0.001$ for time spent), and consequently proved more conjectures ($p=0.004$), particularly in Disjunction, Negation and Quantifiers ($p<0.01$).

Summary of Jape usage by student group

It is surely not implausible that groups such as females, older students, students registered for the single subject computer science degree, students with programming experience, and students expecting the course to be interesting should either be more conscientious or more naturally inclined to try out a new program. This is hardly conclusive evidence that these groups are using Jape because they feel that it specifically addresses their needs. The fact that students who volunteered to take part in the research tend to use Jape more should also be no surprise (perhaps they used the program more because being involved in the research got them interested in using the program; or perhaps they volunteered because they were already interested in using the program).

However, the students who have entry qualifications other than A-levels are also using Jape more, and perhaps the same arguments do not apply for this group.

Another important conclusion from the above analysis is that all too often, in the student groups identified above, the extra time spent on Jape fails to turn convincingly into clearly greater progress (the correlation between usage and progress for all students is 0.73). There may be a number of reasons for this. It is possible that the extra work simply represents greater time being spent on unsuccessful attempts. It is possible that the extra work represents greater distraction from the task. It is possible that these students need the extra time in order to achieve the same progress as the other students. It is possible that these students are being more methodical, are using their notes, are taking notes, and the list could go on.

However, there is also some evidence in the above analysis that the extra progress is on Negation and Quantifiers (and, to a lesser extent, Disjunction) - i.e. the generally harder topics. We have already seen from the analysis of Jape usage by topic that these topics represent somewhat of an impasse for certain students. It is possible, then, that the students who are spending longer on Jape are doing so in order to attempt the harder conjectures. It is striking that the only groups that made extra progress also took extra time.

This can be illustrated by looking at the proportion of students who proved a large number of the conjectures in particular topics. For example, consider Disjunction: Double the proportion of females as males proved 70% of Disjunction (25% compared to 12.5%; χ^2 test: $p=0.032$); and three times the proportion of older students as younger students proved 70% of Disjunction (36% compared to 13.5%; χ^2 test: $p=0.032$). Attempting the harder conjectures - let alone proving them - increases the time that has to be spent.

Finally, a most important and perhaps surprising conclusion that can be drawn from this analysis is that the differences between student groups - even when statistically significant - are very small. If the effects of background variables on Jape usage are minimal, then it would appear that the program can be used by students from a wide variety of backgrounds. One question for further investigation is the reason for this success.

4.5 Survey2 - Feedback about Jape

This section reports the analysis of Survey2. This survey aimed to obtain a snapshot of students' experiences of the course and of Jape. The survey form is in the appendix to this document.

4.5.1 Survey2 - sample

The survey was taken by 66 students taking the logic course. This is roughly 40% of all students associated with the course at some point during the term. The survey took place at the end of the term, shortly before the second test, during a regular lab workshop. However there was a high proportion of absentees from this compulsory workshop.

Three returned surveys were anonymous; the rest of the students were already part of the database of students used in the analysis and virtually all had taken Survey1, Logic1 and Logic2.

Females were 10% more likely to take Survey2 than males. Those with mathematics A-level were 12% more likely to take Survey2 than those without. Other groups identified in Survey1 had similar proportions in Survey2.

4.5.2 Experience of the Course

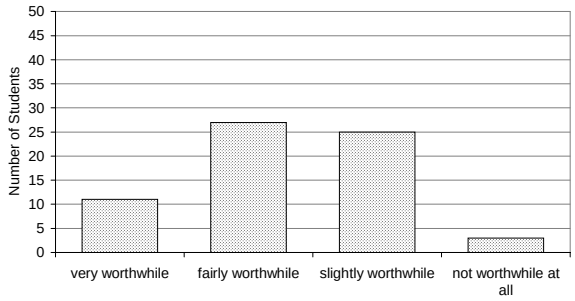


Figure 102 Responses to "How worthwhile was the logic course?"

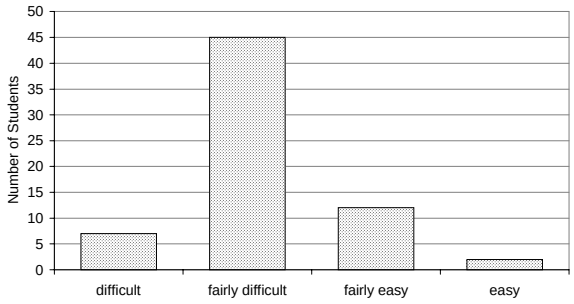


Figure 103: Responses to "How difficult was the logic course?"

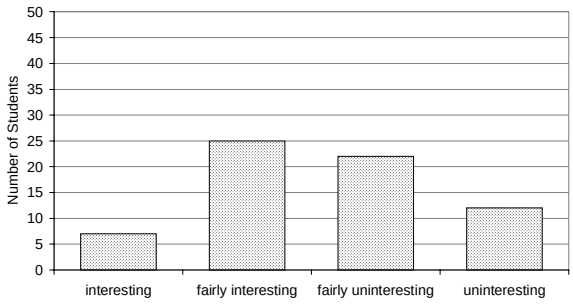


Figure 104: Responses to "How interesting was the logic course?"

Just three students thought the course had not been worthwhile at all. Almost 80% thought the course had been "difficult" or "fairly difficult". Around half thought the course "interesting" or "fairly interesting". About 25% of students said that the course had been (simultaneously) "fairly worthwhile", "fairly difficult" and "fairly interesting".

Students who were mainly motivated to do Computer Science because of an extrinsic reason (for example, because of financial rewards or a stable job) were less likely to consider the logic course worthwhile or interesting, and more likely to consider it difficult.

Students without A-levels tended to consider the course more worthwhile and more interesting than those with A-levels.

The student groups that were more likely than other groups to were consider the course had been "very worthwhile" or "fairly worthwhile" were: students with a computer-related A-level, students without A-levels, students with some programming experience, students for whom "learning to program" was a relevant motive, students who did not indicate "financial rewards" as a relevant motive, and students who indicated "following an interest" as a main motive.

The student groups that were more likely than other groups to were consider the course had been "difficult" or "fairly difficult" were: females (over a quarter of males described the course as "easy" or "fairly easy"; no females did), students registered for the degree "Maths & Computing" (none thought it easy), and students whose main motive was something other than "following an interest". The more difficult or dull students had expected the course to be, the more likely they were to find the course had been difficult or fairly difficult; on the other hand those who had expected the course to be easy or fairly easy tended to find it more difficult than they had expected.

The student groups that were more likely than other groups to were consider the course had been interesting were: students without A-levels, students for whom "learning to program" was a relevant motive. About half of those who had been expecting the course to be interesting or fairly interesting (nearly all the students) indicated afterwards that they thought the course had in fact been either uninteresting or fairly uninteresting.

4.5.3 Time spent using Jape

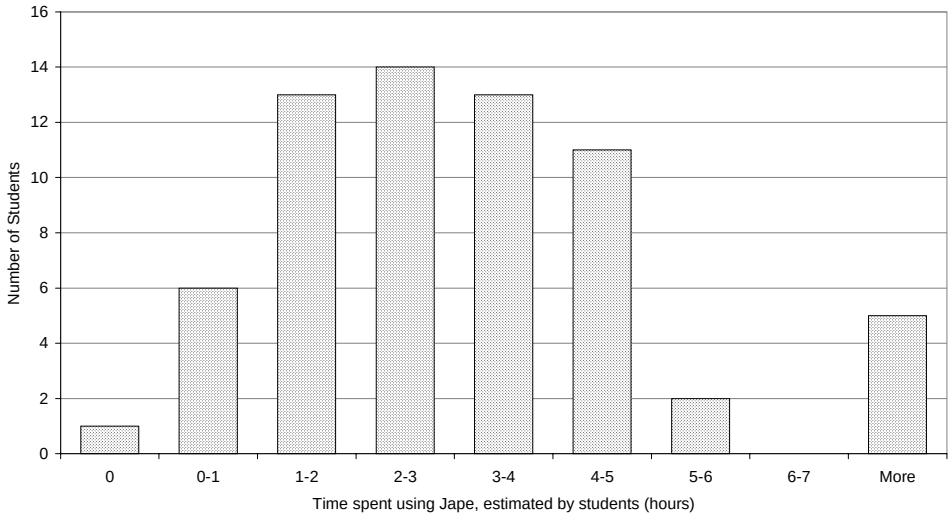


Figure 105: Frequencies of time spent using Jape, as estimated by students

There are 46 students for whom there are estimates of the time spent using Jape from both Survey2 and the logfiles. The student estimates are, in total, almost three times that of the logfile estimates. In only 3 cases were student estimates lower than logfile estimates.

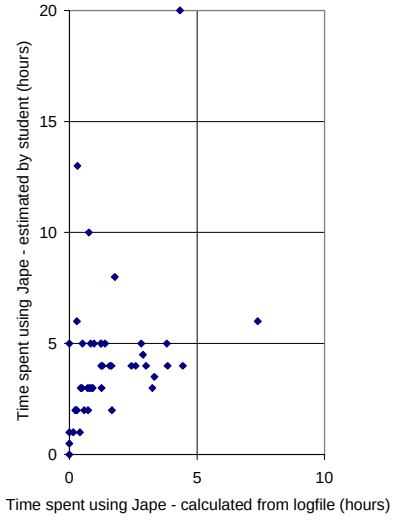


Figure 106: Time spent using Jape, estimated by students and calculated from logfiles

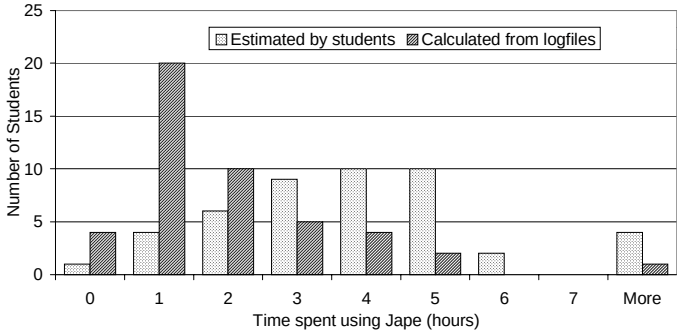


Figure 107: Time spent using Jape, estimated by students and calculated from logfiles

It is of course likely that some students worked with friends or at home, so that logfile estimates would be too low. However, it is also likely that student estimates of "20 hours" and "13 hours" are less than accurate. Once such obvious outliers have been removed (5 in number), the Pearson correlation coefficient is 0.53, and Spearman's rho is 0.64.

4.5.4 Use of Jape's Printing Facility

The students were asked if they had made use of the printing facility within Jape. 63 out of 66 students responded.

8 students (12% of the sample) indicated that they **often** made use of it. 8 students indicated that they **sometimes** used it. 47 students (71%) indicated that they had **never** used it.

4.5.5 Helpfulness of Jape

The students were asked how helpful Jape had been for them overall in learning logic. 65 out of 66 students responded.

2 students (Percy, S64; and Reg, S134; both high-usage students) indicated that they found it **very helpful**. 15 students (23% of the sample) indicated that they found it **fairly helpful**. 30 students (45%) indicated that they found it **slightly helpful**. 16 students (24%) indicated that they found it **not helpful at all**. 2 students indicated that they found it **harmful**. One way of putting this is that 70% of students found Jape helpful.

One of the students who indicated that they found Jape harmful was anonymous and in writing about his dislikes appeared to make reference to finding the idea of working backwards from conclusions difficult. The other student, Omesh (S54) was clearly flippant in most of his answers.

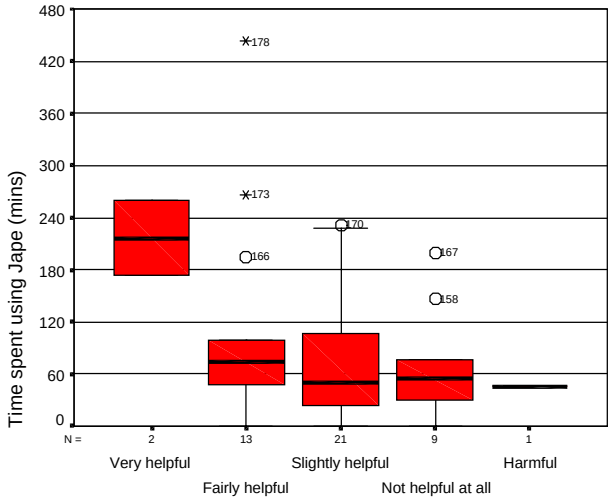


Figure 108: Time spent on Jape, by perceived helpfulness

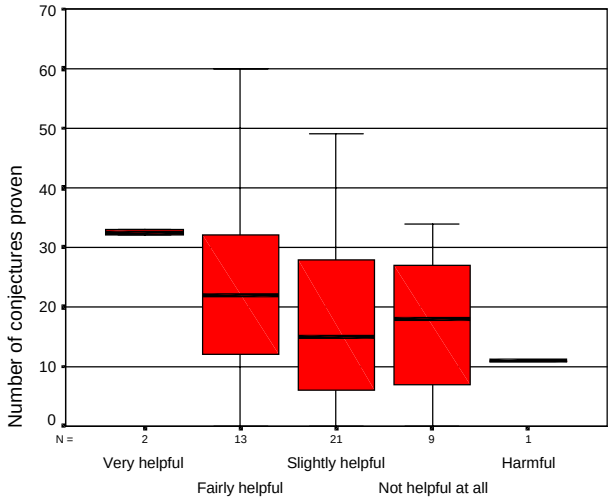


Figure 109: Progress in Jape, by perceived helpfulness

The perceived helpfulness of Jape does not appear to depend strongly on any of the background factors. However, almost none of the students registered for "Maths & Computing" and none of the students who had expected the course to be dull thought that Jape had been "very helpful" or "fairly helpful". Students who indicated "learning to program" as a relevant factor in choosing to study computer science were slightly more lightly (than those hadn't) to consider Jape as "very helpful" or "fairly helpful", but this was still only around a third of them.

4.5.6 Likes about Jape

Students were asked what they most liked about Jape. Of the 66 students, 47 put down at least one thing they liked. However, there was no consensus on *what* they liked.

16 students emphasised the guarantee of correctness provided, in such comments as, "You know when you are right", "You can easily try solutions and get an immediate answer as to whether it is a legal application of the rules", and "You have facility to learn from trial and error".

18 students praised some aspect of the interface, in comments such as, "Having a selection of rules displayed", "Easy to use", and "The interface".

Six students commented on the value of the undo feature: "Easy to backtrack in proofs" and "Can experiment to obtain answer", for example. Eight students explicitly commented on the automatic layout: "Easier to use than drawing out boxes with pen and paper", for example. Eight students mentioned speed as something they liked. Eight students emphasised the opportunity for practice that it provided. Two students said that it was fun.

4.5.7 Dislikes about Jape

Students were asked what they most disliked about Jape. Of the 66 students, 41 put down at least one thing they disliked. Once again, however, there was no consensus on *what*.

28 students made comments relating to the interface. Of these students, 8 referred negatively to trial-and-error was possible; for example, ", "You don't really know what you are doing, i.e. you just keep on clicking buttons 'til you get it right", "You can get the answer too easily by trial & error", "You can click on anything & just guess", "It's like a calculator, type in the question and it gives you the answer. No working out involved.", and "I don't have to think why it's wrong or right, just click".

At least two students apparently referred to when Jape carries out incomplete steps: for example, "Sometimes given very silly answers (when you take wrong step)"and "The way it would introduce its own variables - CONFUSING".

Several students referred to the feedback messages that appear after an attempted illegal step, in comments such as: "Very complicated messages" and "When you make an error, the error message doesn't usually help you.".

Three students commented that some Jape features were not obvious, particularly the hyp command: "Hyp and such functions weren't properly explained and not easy to figure out.".

A few of the interface comments hinted at difficulties with using Jape to learn Natural Deduction: "It kept giving me answers even before I could work out for myself how to do it", "Doesn't list rules explicitly", "Have to do proofs in a specific way which is different to when we learnt", "Defeated the thinking object of logic", and "Sometimes difficult to do what you intended".

Other comments about the interface included: "TERRIBLE, unreadable font used in dialog boxes", "Impossible to print my proof. It changes the symbols e.g. → becomes ∮ ", "Complicated to save what I've done" (referred to a network problem), "Scope boxes in FOL" (supplemented, in the improvements section of the survey, by the comment "Make the scope boxes and their notation more intuitive"), "Lack of a toolbox (i.e. no using commands through menu)".

Two other interesting comments were made. Firstly a couple of students indicated that "It doesn't teach" and that they would like more explanation of the logic (as opposed to the interface). Secondly, one student noted the "Repetitiveness of ready made questions (all are very similar)".

4.5.8 Suggestions for Improvements to Jape

Students were asked how they would improve Jape. Half the students in the sample offered suggestions.

Several students wanted the interface improved, but without specifying in what ways. Typical was "Make it user friendly with understandable messages". Some students suggested sound effects. Another student was in no doubt what he wanted: "Improve the graphical interface. More help files. Change the name.".

Quite a few students asked for more help facilities, a "how to use" guide and more information messages.

One of the students who indicated that Jape had not been helpful wrote, "When a rule is applied that is not helpful then a message should be produced to let you know.". Conversely, some students suggested that correct steps should be acknowledged, although perhaps the suggestion "Some sound effects & acknowledgement of correct answer e.g. flashing screen, dancing women etc." went a little too far.

Julian (S98), who used Jape for about 9 minutes, wrote - with touching honesty - that it could be improved "by me using it more.".

Ken (S162) who mostly attempted just the Implication topic, wrote "Allow more working out options. Don't just give the answers."

The longest suggestion was this one from Noam (S91) - a medium-usage student: "Provide a built-in introductory training course on how to use the software as well as learning the language of logic. It should be done with an interactive approach. Also error messages should be simpler to understand and better help facilities should be provided."

Ronald (S93), a medium-usage student, wrote "Improve error messages. Make it so that trial and error does not lead to correct answer. Make it so you can use it on Windows NT, 95 or 98 alongside Linux. Thanks."

One of the students who indicated that Jape had not been helpful (Stan, S55), when asked how he would improve Jape wrote “I wouldn’t - it's just not for me but it's fine.”. He had used Jape for 76 minutes.

Donald (S89), who used Jape for about 3½ hours, suggested “Maybe a window on it to give the definition of a rule at the time that we choose a rule to apply.”.

Marcel (S94), a high-usage student, suggested “Make it easier to perform ∀ and ∃ questions (i.e. remove complexity of underlying variables)”.

Alan (S118), a high-usage student, suggested “Make quick key references to all boolean operators, like Ctrl-E etc.”

4.6 Logic1 - First Written Test

This section gives a brief overview of the analysis of the first logic test (“Logic1”) that the students took. The test is primarily on propositional logic, and about a third of the test involves constructing natural deduction proofs.

Throughout this section, t-tests should be assumed to be with equal variances, unless otherwise stated (in which case a Levene test will have suggested this is an inappropriate assumption).

4.6.1 Logic1 - sample

The test was taken by 163 students on the logic course. This is 92% of all students associated with the course at some point during the term.

The test took place halfway through the course. It was a compulsory part of the course, so those who were missing had, for example, dropped out of the course or were ill.

For students who took both Logic1 and Survey1, the proportions of the various groups identified in Survey1 were similar in both the test and the survey. However the mean score in Logic1 for those who had taken Survey1 was higher than for those who had missed it (62% for 138 students compared to 45% for 25 students; $p<0.001$ for a 2-tailed t-test). Setting aside the explanation that Survey1 assisted with Logic1 in some way, it could well be that those who were absent for Survey1 either started the course late or were more likely to miss lab workshop, lectures and tutorials.

Similarly, the mean score in Logic1 for those who had taken the mathematics test was higher than for those who had missed it (62% for 119 students compared to 52% for 44 students; $p=0.009$ for a 2-tailed t-test).

For the purposes of this analysis, the data for students who were not present for Survey1 has been sought - in particular the data on degree course, gender, age, and A-levels.

4.6.2 Logic1 - % score

The mean score was 59%, with standard deviation 20.7.

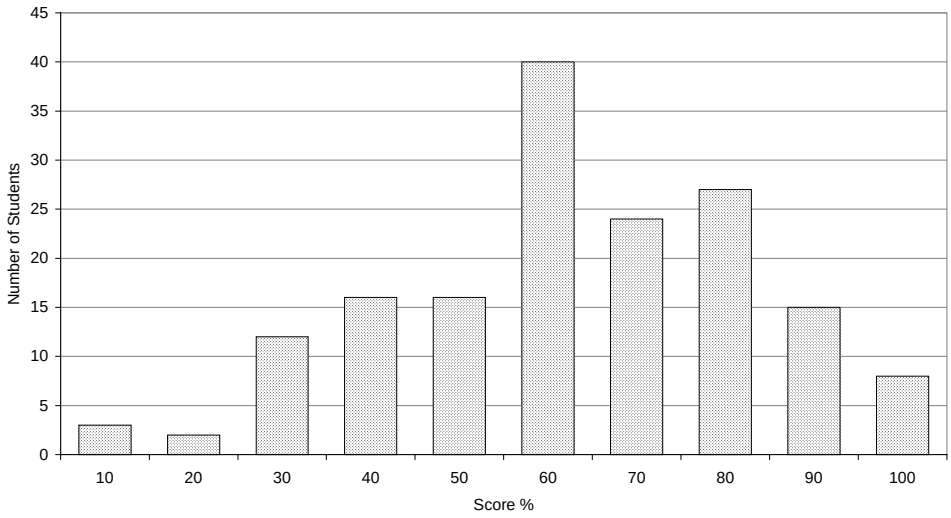


Figure 110: Student Logic1 scores

4.6.3 Logic1 - scores by student group

The mean and standard deviation of the test are virtually identical to those of the mathematics test, however there appears to be little correlation between the scores (Peason=0.258). ANOVA suggests that Survey1 does not contain good predictors for Logic1. There are no significant differences in mean scores between most of the student groups identified in Survey1.

The mean score for females is higher than for males (66% compared to 58%; p=0.016 for a 1-tailed t-test).

The mean score for students registered for the degree "Computer Science & Mathematics" is higher than for those registered for "Computer Science" (p=0.003 for a 1-tailed t-test) and than for those registered for "Computer Science & Business Studies" (p=0.013). The mean score for students registered for the degree "Mathematics & Computing" is *lower* than for those registered for "Computer Science" (p<0.001) and than for those registered for "Computer Science & Business Studies" (p=0.003).

Degree	N	Mean Score	σ
Computer Science & Mathematics	15	76%	16.1
Computer Science	100	61%	19.5
Computer Science & Business Studies	20	61%	19.0
Mathematics & Computing	21	44%	20.4
Linguistics & Computer Science	2	41%	1.4
Mathematics	1	86%	-
Computer Systems & Digital Electronics	3	34%	6.0
Overall	162	59%	20.7

Figure 111: Student Logic1 scores, by registered degree

The following results seem to suggest that the harder the logic course was expected to be, the worse the score. However, differences between the groups are not statistically significant.

Expectation	N	Mean Score	σ
difficult	18	55%	22.1
fairly difficult	77	62%	18.3
fairly easy	40	64%	20.3
easy	3	74%	12.2
Total	138	62%	19.4

Figure 112: Student Logic1 scores, by expectations of the course

4.7 Logic2 - Second Written Test

This section gives a brief overview of the analysis of the second logic test ("Logic2") that the students took. The test is primarily on predicate logic, and about a third of the test involves constructing natural deduction proofs.

4.7.1 Logic2 - sample

The test was taken by 144 students on the logic course. This is roughly 80% of all students associated with the course at some point during the term.

149 students completed Survey1, and of these 149, 110 took the maths test, 138 took Logic1 and 124 took Logic2. 91 students completed all four instruments. The two students for whom there is the most complete video record of their work with Jape during the Observational Study are among this 91.

The test took place at the end of the term-long course. It was a compulsory part of the course, so those who were missing had, for example, dropped out of the course or were ill.

Of those who took Logic1, 23 students did not take Logic2. Their scores in Logic1 were representative of all who took the test. Of those who took Logic2, 4 students had not taken Logic1. They scored well below average on Logic2.

Of those who completed Survey1, 22 students did not take Logic2. It is interesting that of the 38 students who indicated that their main motivation for doing Computer Science was "financial rewards", over a quarter of them failed to take Logic2. For comparison, of the 52 students who indicated that their main motivation for doing Computer Science was "following an interest", just 6% of them failed to take Logic2. This difference in proportions is significant ($p=0.003$ for a 1-tailed t-test).

There may also be a small effect whereby those who expected the logic course to be interesting were more likely to miss Logic2 than those who did not expect it to be interesting. However this difference is not significant at the 1% level. (6% of those who expected an interesting course missed Logic2, compared to 18% of the rest; $p=0.020$ for a 1-tailed t-test).

Other various groups identified in Survey1 had similar proportions in Logic2 and Survey1.

20 students took Logic2 but did not complete Survey1. Their mean score was significantly lower than the score for those who took both (37% for 20 students compared to 58% for 124 students; $p<0.001$ for a 1-tailed t-test). Just as for Logic1, an explanation for this could be that those who were absent for Survey1 either started the course late or were more likely to have or missed lab workshop, lectures and tutorials.

4.7.2 Logic2 - % Score

The mean score for Logic2 was 55%, with standard deviation 22.7.

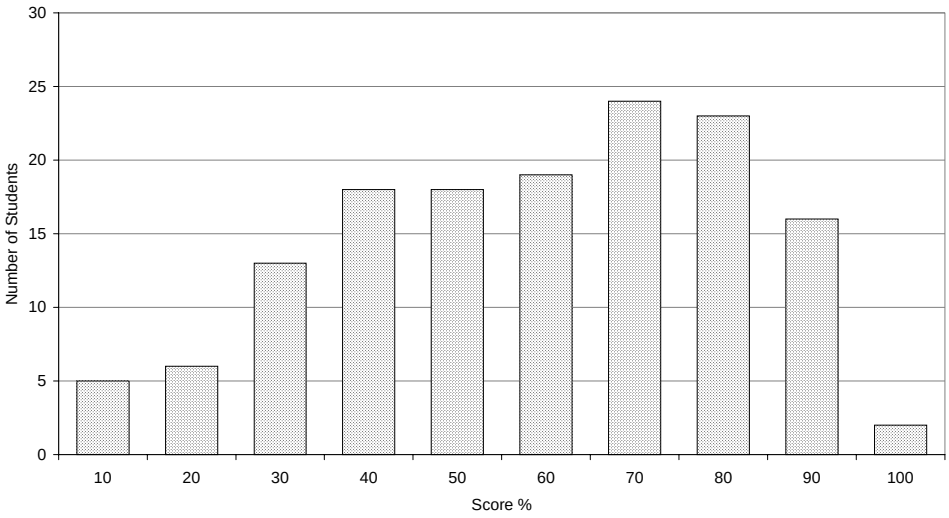


Figure 113: Student Logic2 scores

4.7.3 Comparison between Logic1 and Logic2

The mean for Logic2 is lower than for Logic1 (55% compared to 59%; p=0.001 for a 1-tailed, paired t-test).

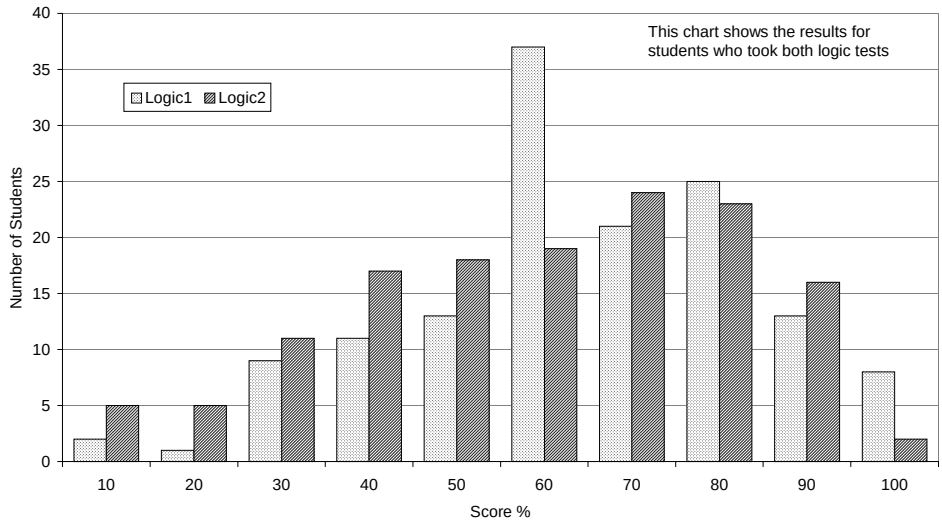


Figure 114: Frequencies of student scores for Logic1 and Logic2

The correlation between the tests is only 0.55:

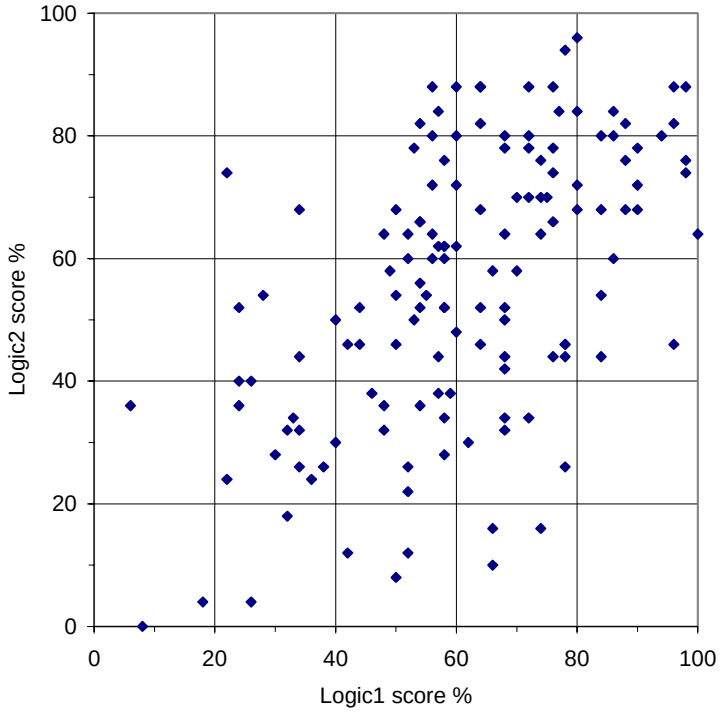


Figure 115: Comparison of Logic1 and Logic2 Scores for each student

4.7.4 Logic2 - scores by student group

The scores for Logic2 are not correlated with those for the mathematics test (Peason=0.258). Linear regression suggests that Survey1 does not contain good predictors for Logic2. Just as was the case with Logic1, there are no significant differences in Logic2 mean scores for many of the student groups identified in Survey1.

However, the mean score for females is again higher than for males (64% compared to 53%; p=0.010 for a 1-tailed t-test), and there are four interesting results for Logic2 that were not found in Logic1: for the financial motive, for age, for the registered degree and for home computer usage.

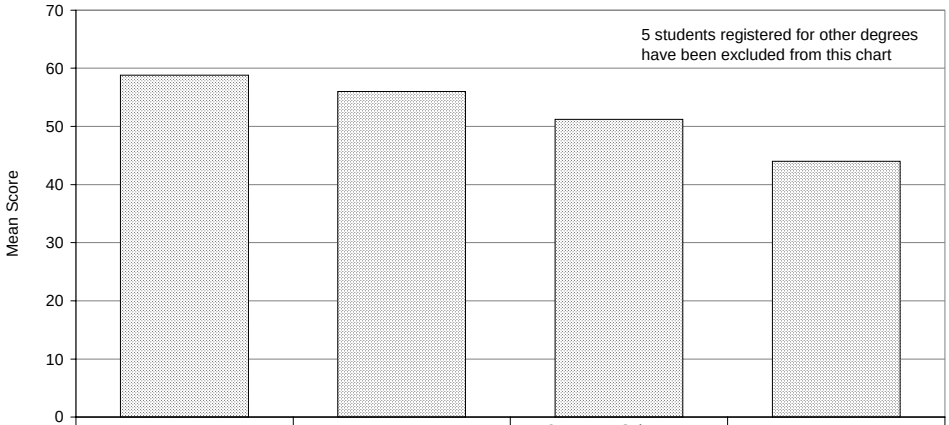
The mean score in Logic2 for those who indicated that "financial rewards" was a *relevant* factor (i.e. not necessarily the most important factor) in their decision to study Computer Science is lower than for those who didn't indicate it (55% compared to 65%; p=0.018). Similarly, even though (as has already been noted) a quarter of those indicating the financial motive as the *most important* factor failed to even take Logic2, those that *did* take the test fared worse than those who indicated another factor as most important (51% compared to 60%; p=0.021).

It might be that this finding explains the better results for females, in that a smaller proportion of females than males indicated the financial motive as the *most important* factor (18% compared to 27%, among those who took Logic2) or as a *relevant* factor (61% compared to 80%). It is, of course, also possible in principle that the explanation is the other way round - i.e. that gender explains the worse results for those with financial motivation.

The mean score for the 9 older students who took Logic2 is higher than for the 126 younger students (72% compared to 55%; p=0.013 for a 1-tailed t-test). Such a large difference was not found in Logic1.

However, the difference might again be explainable in terms of the financial motive - none of the older students indicated it as the most important factor.

The mean score for the 84 students registered for the “Computer Science” degree is higher than for the 21 students registered for “Mathematics & Computing” (59% compared to 44%; p=0.003 for a 1-tailed t-test). This was found in Logic1. However, the mean differences between “Computer Science & Mathematics”, “Computer Science & Business Studies” and “Mathematics & Computing” are no longer significant, as they were in Logic1.



	Computer Science	Computer Science & Maths	Computer Science & Business	Maths & Computer Science
Mean	59	56	51	44
N	84	15	18	21

Figure 116: Logic2 scores, by registered degree

The financial motive may once again be an explanatory factor in the relative success of students registered for “Computer Science”. The proportion of students registered for “Computer Science” who chose “financial rewards” as the most important factor (17%) was smaller than for those registered for “Computer Science & Maths” (27%), for “Computer Science & Business” (53%) or for “Maths & Computer Science” (33%).

There are some odd results for home computer usage that were not shown for Logic1. For example, although there is little difference in score between students who have had use of a computer at home before they started the course and the others, and the difference in mean score between students with prior programming experience and the others is not significant, students who have programmed at home did slightly better than those who didn’t (63% compared to 56%; p=0.030 for a 1-tailed t-test); whereas students who have used spreadsheets at home did slightly *worse* than those who didn’t (55% compared to 62%; p=0.032 for a 1-tailed t-test).

The financial motive is again pertinent here. 95% of those who indicated it as the most important factor had little or no programming experience. This compares with 67% of those who did not indicate it as the most important factor. Moreover, of those who have programmed at home, 18% indicated the financial motive as being the most important factor, compared to 34% of those who have not programmed at home.

4.8 Exam - Third Written Test

This section gives a brief overview of the analysis of the logic exam that the students took some 5 months after the end of the logic course.

4.8.1 Exam - sample

The exam was taken by 145 students on the logic course. This is roughly 80% of all students associated with the course at some point during the term. 7 students were absent, and 23 students had left the course for one reason or another.

88% of these students had completed Survey1; 99% had taken Logic1; 90% had taken Logic2.

18 students took the exam but did not complete Survey1. Their mean score was slightly lower than the score for those who took both (37% for 18 students compared to 46% for 127 students; $p<0.014$ for a 1-tailed t-test). Just as for the logic tests, an explanation for this could be that those who were absent for Survey1 either started the course late or were more likely to have or missed lab workshop, lectures and tutorials.

Of those who completed Survey1, 19 students did not take the exam. 14 left the course and 5 were absent.

Those without mathematics A-level seem to be a special group: 21% of them had left the course before the exam (compared with under 10% of those *with* mathematics A-level); and 14% of them were absent from the exam (compared with 2% of those *with* mathematics A-level).

The proportions of other groups identified in Survey1 showed little difference between Survey1 and the exam.

4.8.2 Exam - % score

The mean score for the exam was 56%, with standard deviation 19.6.

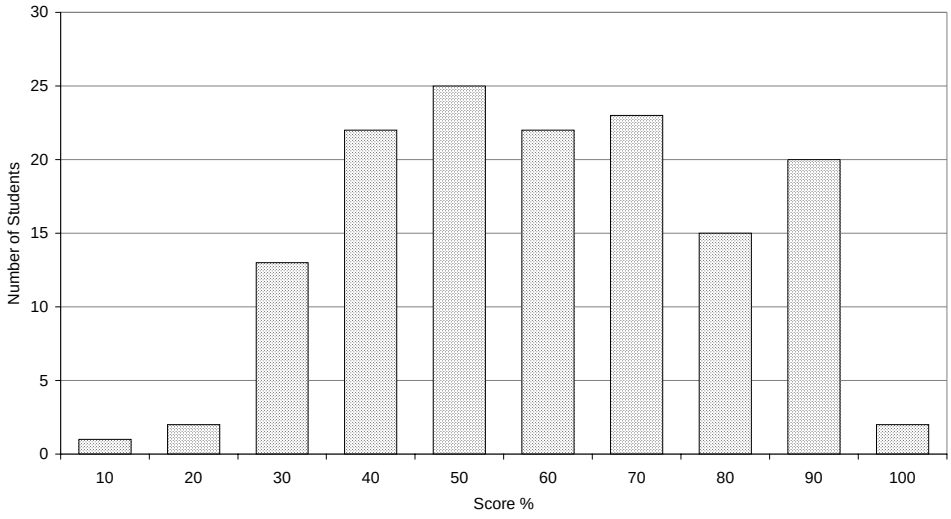


Figure 117: Student Exam scores

4.8.3 Comparison between Exam and other tests

The mean for the exam is about the same as for Logic2, and slightly lower than Logic1 ($p<0.001$ for a 1-tailed paired t-test). The correlation between the exam and Logic1 is 0.43; the correlation between the exam and Logic2 is 0.62. However, although the exam contained questions on both propositional logic (the topic for Logic1) and predicate logic (the topic for Logic2), students only had to answer 4 questions out of 6, and so could in principle leave out both the natural deduction questions.

4.8.4 Exam - by student group

Linear regression suggests that Survey1 does not contain good predictors for the exam. Just as was the case with the logic tests, there are no significant differences in mean exam scores for many of the student groups identified in Survey1. The scores for the exam are not correlated with those for the mathematics test (Peason=0.25).

The mean score for females is higher than for males (62% compared to 54%; $p=0.012$ for a 1-tailed t-test). This is similar to the logic tests.

There may be a slight advantage for those who have experience of programming at home, but it is not significant at the 1% level (62% for 38 students, compared to 55% for 88 students; $p=0.022$ for a 1-tailed t-test). The difference was significant in Logic2, but not in Logic1. There may also be a slight advantage for those with “a fair amount” or

"a lot" of prior programming experience (64% for 29 students, compared to 55% for 97 students; p=0.015 for a 1-tailed t-test). This difference was not significant in either Logic1 or Logic2. There was no significant disadvantage for those with experience of spreadsheets at home, as there was in Logic2.

The mean exam score for those who indicated that "financial rewards" was a *relevant* factor in their decision to study Computer Science is lower than for those who didn't indicate it (54% compared to 65%; p=0.003). The difference was significant in Logic2, but not in Logic1.

The mean exam score for those who indicated "financial rewards" as the *most important factor* was not significantly lower than for the rest (54% compared to 58%). The difference was significant in Logic2, but not in Logic1.

The mean exam score for those registered for "Computer Science" degree is higher than for those registered for "Mathematics & Computing" (like both Logic1 and Logic2; 58% for 84 students, compared to 46% for 20 students; p=0.005). The mean score for those registered for the "Computer Science & Mathematics" degree is higher than for those registered for "Mathematics & Computing" (like Logic1, but unlike Logic2; 62% for 15 students, compared to 46% for 20 students; p=0.011 for a 1-tailed t-test). The mean score for those registered for "Computer Science" is not significantly different from that for those registered for "Computer Science & Mathematics" (like Logic2, but unlike Logic1).

The mean exam score for those who expected the logic course to be "difficult" or "fairly difficult" is slightly lower than those who expected it to be "easy" or "fairly easy" (62% for 40 students, compared to 55% for 87 students; p=0.021). The difference was not found significant in Logic1 or Logic2.

4.9 Analysis of Factors Influencing Outcomes

This section gives an overview of the analysis of the factors - background factors and Jape usage factors - that might have influenced the scores obtained by students in their tests and exams.

In this section, t-tests should be assumed to be with equal variances, unless otherwise stated (in which case a Levene test will have suggested this is an inappropriate assumption).

4.9.1 Sample

Based on earlier analysis, it has been found that there are 122 students who used Jape, 24 students who didn't use Jape (at least not using their college computer account), and 32 students of unknown status. When Jape usage is considered in this section, the students of unknown status will be excluded from the analysis, leaving 146 students whose Jape usage is known.

163 students took Logic1, 12 did not, and 3 are of unknown status. 144 students took Logic2, 31 did not, and 3 are of unknown status. 145 students took the exam, 30 did not (23 of whom had left the course, and 7 were absent), and 3 are of unknown status. When test scores are considered in this section, the students of unknown status will be excluded from the analysis, leaving 175 students whose test status is known.

129 students took each of Logic1, Logic2 and the exam; 46 students missed one or more tests. Of the 46 students who missed one or more tests, 23 dropped out of the course - they withdrew (or were withdrawn), or they transferred to another course, or they transferred to another university, or they simply failed the course. Of these 23, none took the exam, 7 took both Logic1 and Logic2, 8 missed one of either Logic1 or Logic2, and 8 missed both Logic1 and Logic2. Of the remaining 23 students - those who missed at least one test but didn't drop out of the course - 4 students missed just the exam, 2 students missed just Logic1, 14 students missed just Logic2, and 3 students missed Logic2 and the exam.

In calculating the overall mark for the course, the formula used by the college is that Logic1 counts for 10%, Logic2 counts for 10%, and the exam counts for 80%. However, students close to the grade boundaries may have their marks adjusted when other factors are taken into account.

Grade	Boundary	N
A	70	36
B	60	27
C	50	24
D	45	13
E	40	22
F		30
absent		7
Total		159

Figure 118: Number of students obtaining each grade for the logic course

4.9.2 Background Factors

For the overall course mark, the mean score of the 31 females was higher than for the 110 males (63% compared to 54%; p=0.008 for a 1-tailed t-test). The same result holds for Logic1, Logic2 and the exam (at the 5% level for each).

For the overall course mark, the mean score of the 20 students registered for the degree "Mathematics & Computing" was lower than for the 124 others (46% compared to 57%; p=0.005 for a 1-tailed t-test). The same result holds for Logic1, Logic2 and the exam (at the 5% level for each).

For the overall course mark, the mean score of the 55 students with no prior programming experience was lower than for the others (53% compared to 61%; p=0.007 for a 1-tailed t-test). The same result holds for the exam (at the 1% level), but the difference was not significant for Logic1 and Logic2. The gap in average score for the exam between those who indicated "a fair amount" of programming experience and those who indicated "none at all" was 15%.

For the overall course mark, the mean score of the 96 students who indicated that financial rewards were relevant in their choosing to study a degree that included computer science was lower than for the 31 others (55% compared to 65%; p=0.003 for a 1-tailed t-test). The same result holds for Logic2 (at the 5% level) and the exam (at the 1% level).

For each of these four factors - gender, degree, programming experience and financial motivation - the student mortality from Logic1 to the exam was around 10% (with the exception of those without prior programming experience, which lost 5% of its students between the two tests).

No consistent differences in test scores were noticed for participants in the research, older students, students with particular prior formal qualifications, students from overseas, or students with differing expectations of the course.

The proportion of students indicating that the course was very worthwhile or fairly worthwhile was 79% for students with a computer at home and 48% for the others (χ^2 test: p=0.011); 68% for students for whom learning to program was a motivation and 41% for the others (χ^2 test: p=0.026); 50% for students who thought the course difficult or fairly difficult and 85% for the others (χ^2 test: p=0.012); and 91% for students who thought the course interesting or fairly interesting and 23% for the others (χ^2 test: p<0.001).

The proportion of students indicating that the course was difficult or fairly difficult was 72% for males and 100% for females (χ^2 test: p=0.009); 58% for students who have programmed at home and 88% for the others (χ^2 test: p=0.005); 65% for students whose main motive was following an interest and 93% for the others (χ^2 test: p=0.008); all 14 of those who expected the course to be uninteresting or fairly uninteresting, and 71% of the others; and 69% for students who thought the course worthwhile or fairly worthwhile and 93% for the others (χ^2 test: p=0.013).

The proportion of students indicating that the course was interesting or fairly interesting was 62% for students for whom learning to program was a motivation and 32% for the others (χ^2 test: p=0.012); 81% for students who thought the course very worthwhile or fairly worthwhile and 11% for the others (χ^2 test: p<0.001); and 44% for students who thought the course difficult or fairly difficult and 77% for the others (χ^2 test: p=0.017).

4.9.3 Use of Jape at least once

For the overall course mark (also known as the "ItL mark"), the mean score of those who used Jape was higher than for those who did not use Jape (58% compared to 42%, p=0.001 for a 1-tailed t-test). This result applies for each of the component tests (Logic1, Logic2 and the exam).

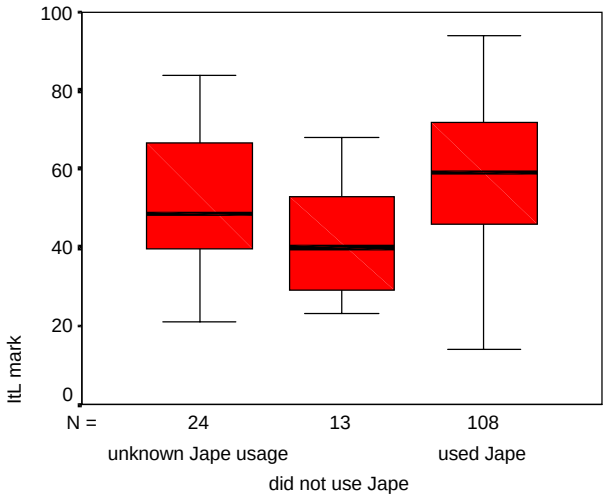


Figure 119: Overall course mark, for Jape users and non-users

It is of course possible either that Jape enables better scores, or that high scorers would be more likely to use Jape, or that another factor is responsible (such as conscientiousness). Moreover there are only 13 exam students who did not use Jape, and Jape usage was not properly recorded for "Maths & Computing" students (hence the high numbers in the "unknown" category).

However, the mean difference in score for Jape users versus non-Jape users is 16%. This is higher than for degree (11%), financial motivation (10%), gender (9%) and programming experience (8%). This is good prima facie evidence that use of Jape is associated with success.

4.9.4 Different levels of Jape usage

The following chart shows that just using Jape for a long time is neither necessary nor sufficient for a good score; however all but one of the high-usage students passed and all but three of the medium-usage students passed.

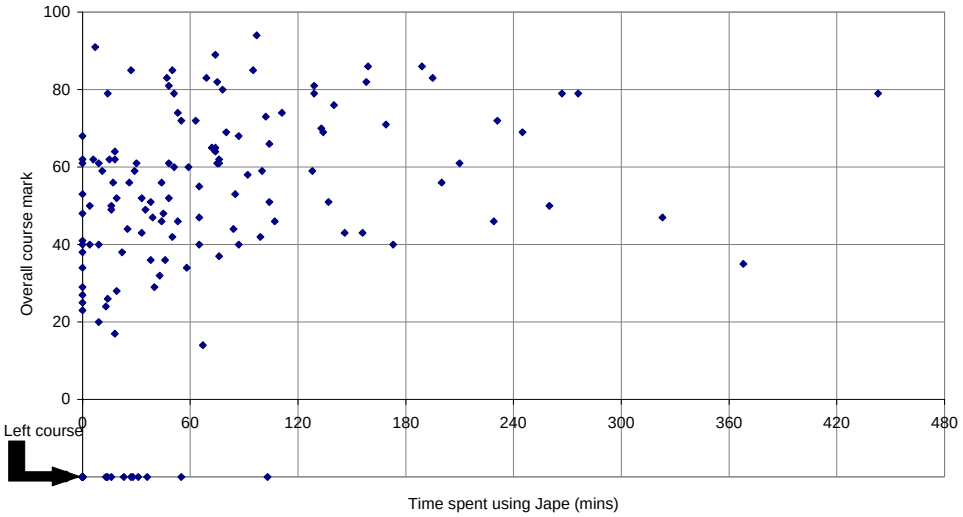


Figure 120: Overall course mark against time spent using Jape

The following table shows the mean % test scores for each of the three written tests, by level of Jape usage:

	N			Mean			Std. Deviation		
	Logic1	Logic2	Exam	Logic1	Logic2	Exam	Logic1	Logic2	Exam
Zero	19	13	13	48	40	42	16.5	20.6	15.4
Low	58	49	51	57	52	53	20.4	20.3	19.1
Medium	32	29	31	71	56	61	15.4	23.3	19.6
High	28	27	26	69	70	64	14.7	18.5	18.1

Figure 121: Mean test scores, by level of Jape usage (time spent proving)

For each test, the mean score of low-usage students is greater than for zero-usage students (at the 5% level in each case). Moreover, the mean differences between high-usage and low-usage students are significant for each test (at the 1% level in each case, with differences in mean test score of between 10 and 20%).

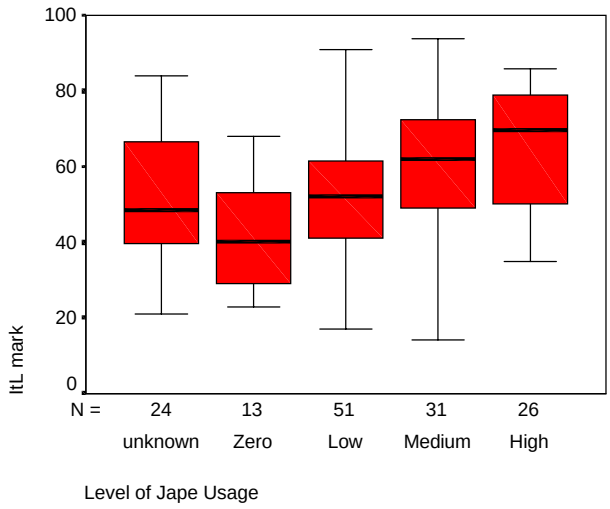


Figure 122: Overall course mark, by level of Jape usage

There are disparities between the tests. The mean score of medium-usage students is greater than for low-usage students overall (a 9% advantage, p=0.019), in Logic1 (a 15% advantage, p<0.001) and in the exam (an 8% advantage, p=0.030). However, the 3.5% advantage in Logic2 is not significant at the 5% level. Also, the mean score of high-usage students is significantly greater than for medium-usage students only in Logic2 (70% compared to 56%, p=0.007).

If Jape is causing an increase in scores, these disparities between the tests suggest that even low-usage makes a difference, that medium-usage is required to make a difference to Logic1, and that high-usage is required to make a difference to Logic2. In explaining why high-usage students do not do significantly better than medium-usage students in the exam, it is possible that the extra advantage offered by being a high-usage student compared to being a medium-usage student fades over time. However, it is more plausible that the exam dilutes any advantage because of the mix of questions. The natural deduction questions in Logic1 were from the Implication, Conjunction and Disjunction topics; those in Logic2 were from the Negation and Quantifier topics. The exam contained a mix of these, arguably skewed towards the easier end of the scale with regard to the Negation and Quantifier topics.

There are differences in levels of Jape usage between the 28 students who missed the Logic2 test and the 118 students who took the Logic2 test (χ^2 test: p=0.001). There are no statistically discernible differences between those who missed and took the other tests.

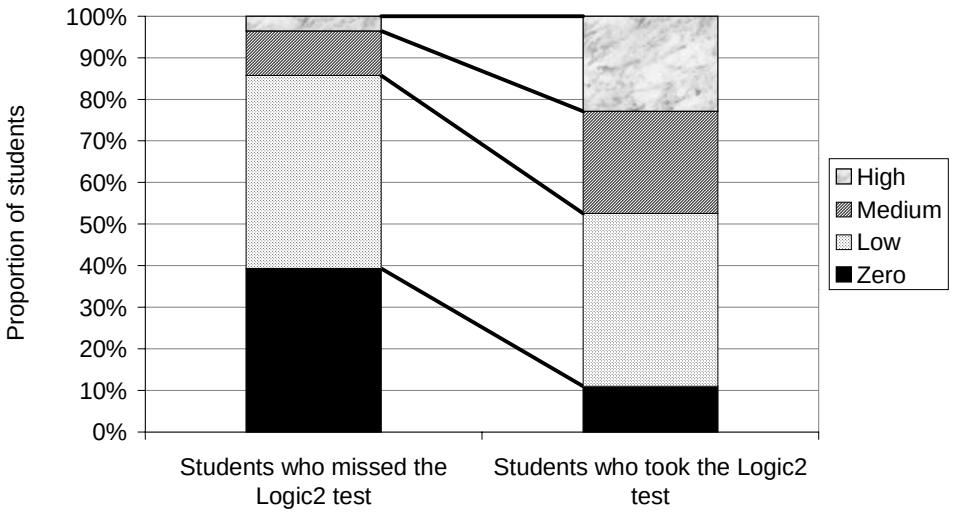


Figure 123: Levels of Jape usage, for students who missed and took Logic2

None of the medium-usage or high-usage students failed Logic1 or Logic2. The 101 students who passed the course overall were heavier users of Jape than the 39 students who failed (χ^2 test: p<0.001).

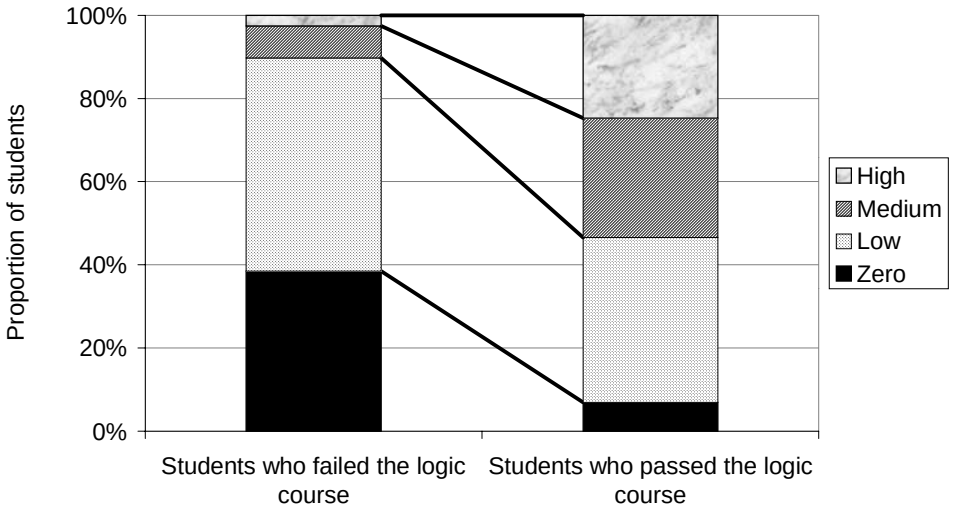


Figure 124: Levels of Jape usage, for students who passed and failed the course

Of the students who were given Survey2 and whose Jape usage was recorded, all 12 high-usage students thought that the course was at least slightly worthwhile. The 3 students who thought that the course was not worthwhile were low-usage students.

There were slight differences in Jape usage between those found the course interesting and those who did not:

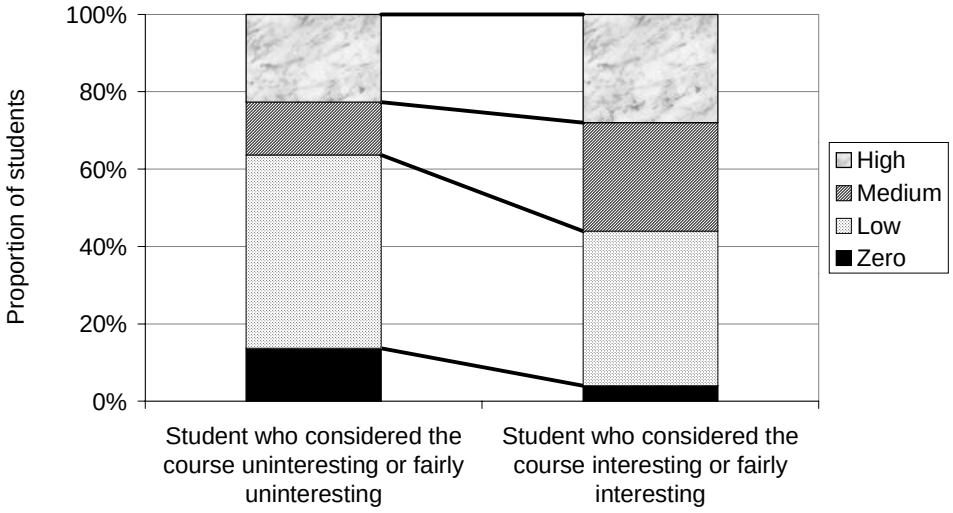


Figure 125: Levels of Jape usage, by interest in the course

4.9.5 Different levels of Jape progress

The following chart shows how the amount of progress made in Jape relates to success in the tests.

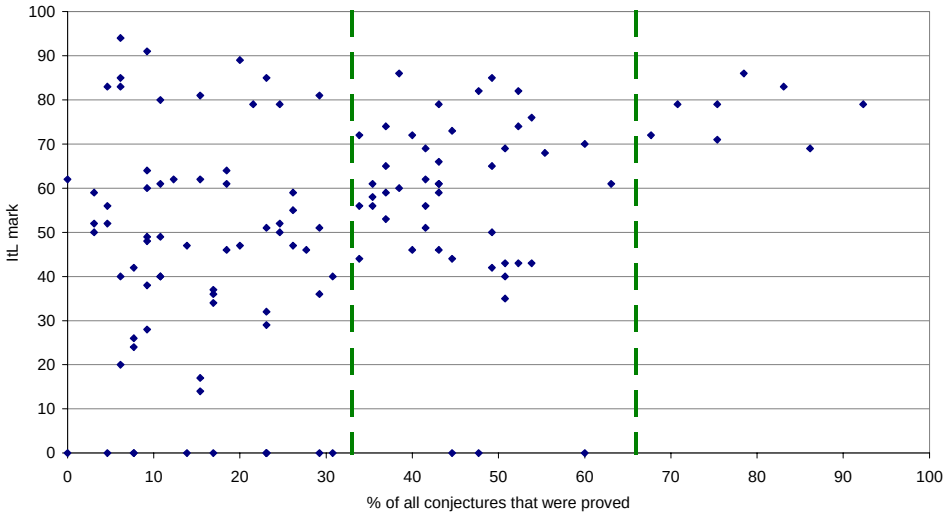


Figure 126: Overall course mark against progress made with Jape

The following table shows the mean % test scores for each of the three written tests, by level of Jape progress:

	N			Mean			Std. Deviation		
	Logic1	Logic2	Exam	Logic1	Logic2	Exam	Logic1	Logic2	Exam
Zero	21	15	14	46	42	43	18.4	20.9	15.7
Low	62	54	56	59	53	53	19.6	23.1	21.7
Medium	46	41	43	67	59	61	15.2	18.6	15.3
High	8	8	8	80	84	76	14.1	5.9	7.7

Figure 127: Mean test scores, by level of Jape progress (number of conjectures proved)

The mean score of high-progress students is greater than for low-progress students for all tests (at the 1% in each case), with differences in mean test score of at least 20%. The mean score of high-progress students is also greater than for medium-progress students for all tests (at the 5% level for Logic1, and at the 1% for Logic2 and the exam), with differences in mean test score of 13-24%.

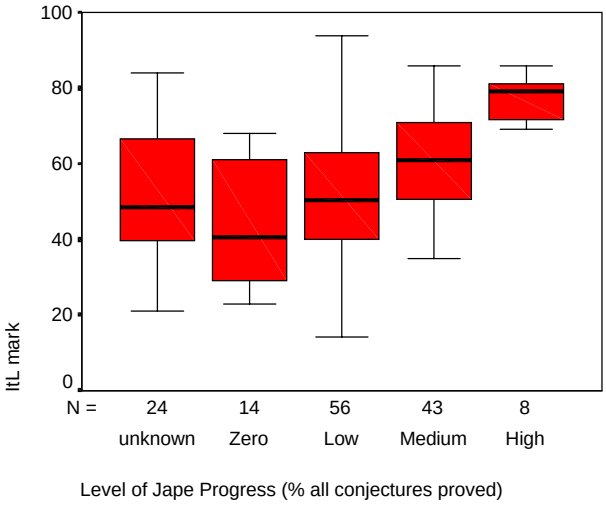


Figure 128: Overall course mark, by level of Jape progress

There are disparities between the test. The mean score of low-progress students is greater than for zero-progress students for Logic1 (a difference of 13%, $p<0.01$) and for Logic2 (a difference of 11%, $p<0.05$); but the difference for the exam (10%) is not significant. The mean score of medium-progress students is greater than for low-progress students for Logic1 (a difference of 8%, $p<0.05$) and the exam (a difference of 8%, $p<0.05$); but the difference for Logic2 (6%) is not significant.

These disparities between the tests are possibly attributable to small group sizes, but it also noticeable that there is a large range of scores by students who made low progress. There is a possible explanation for this - some students indicated during the fieldwork that they had decided to study using their lectures notes rather than persevering with Jape.

All 8 students who made high-level progress with Jape passed the course (7 with grade A, 1 with grade B). All but 2 students who failed the course made low-level progress with Jape.

Out of all students for whom Jape usage figures are available (whether they used Jape or not), the mean score of the 77 students who completed at least 70% of the *Implication* conjectures is about 10% higher than that of the 69 students who did not (significant at the 1% level). Similarly, the mean score of the 58 students who completed at least 70% of the *Conjunction* conjectures is about 10% higher than that of the 87 students who did not (significant at the 1% level). Again, the mean score of the 23 students who completed at least 70% of the *Disjunction* conjectures is about 10% higher than that of the 142 students who did not (significant at the 1% level). For Implication and Conjunction, there is a decline in advantage from Logic1 to Logic2 to the exam (from 15% to 10% to 8% for Implication; and from 13% to 9% to 8% for Conjunction). For Disjunction, however, the advantage for Logic1 is 10%, the advantage for Logic2 is 15%, and the advantage for the exam is 11%. This suggests that the greatest advantage for Logic2 comes from having completed more of Disjunction. Only 3 students completed more than 70% of Negation, and only 2 students completed more than 70% of Quantifiers. Their overall test scores lay in the top ten in the year group.

Of the students who were given Survey2 and whose Jape usage was recorded, all 5 high-progress students thought that the course was at least slightly worthwhile. The 3 students who thought that the course was not worthwhile were low-progress students.

The proportion of students indicating that the course was very worthwhile or fairly worthwhile was 69% for students who had proved at least 70% of Implication and 45% for the others (χ^2 test: $p=0.030$).

The proportion of students indicating that the course was interesting or fairly interesting was 75% for students who had found Jape very helpful or fairly helpful and 41% for the others (χ^2 test: $p=0.010$).

The proportion of students indicating that Jape had been very helpful or fairly helpful was 36% for students who had expected to find the course interesting or fairly interesting and *none* of the others; 38% for students who had

proved at least 70% of Implication and 13% for the others (χ^2 test: p=0.015); 39% for students who had proved at least 70% of Conjunction and 17% for the others (χ^2 test: p=0.026); 60% for students who had proved at least 70% of Disjunction and 19% for the others (χ^2 test: p=0.004); 39% for students who had found the course interesting or fairly interesting and 13% for the others.

4.9.6 Comparison of Background and Jape Factors

Using the four significant background factors, the following ANOVA table for overall course mark is produced:

Tests of Between-Subjects Effects

Dependent Variable: ItL mark

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	7411.462[a]	10	741.146	2.912	.003
Intercept	98046.667	1	98046.667	385.186	.000
BSEX	1012.625	1	1012.625	3.978	.048
BDEGREE3	1380.327	1	1380.327	5.423	.022
BPEXERS	1907.566	1	1907.566	7.494	.007
BMOTIVE3	420.491	1	420.491	1.652	.201
BSEX * BDEGREE3	44.213	1	44.213	.174	.678
BSEX * BPEXERS	261.007	1	261.007	1.025	.313
BSEX * BMOTIVE3	24.815	1	24.815	.097	.755
BDEGREE3 * BPEXERS	94.494	1	94.494	.371	.544
BDEGREE3 * BMOTIVE3	31.174	1	31.174	.122	.727
BPEXERS * BMOTIVE3	324.768	1	324.768	1.276	.261
Error	29272.538	115	254.544		
Total	455690.000	126			
Corrected Total	36684.000	125			

a. R Squared = .202 (Adjusted R Squared = .133)

BSEX: 1=male, 0=female

BDEGREE3: Degree course (1=Maths & Computing, 0=other)

BMOTIVE3: Financial rewards are relevant to the decision to study computer science (1=yes, 0=no)

BPEXRS: Programming experience (1=some, 0=none)

Figure 129: ANOVA for background factors

So it seems as though programming experience is significant at the 1% level, gender and degree course are significant at the 5% level, and financial motivation is not significant

In considering the effect of Jape, there is a problem: Because of the small numbers of students who failed to use Jape but also have an overall course score, ANOVA is not feasible with a simple yes/no comparison of Jape usage. Moreover, since Jape usage data is unfortunately only available for 3 “Maths & Computing” students (2 of whom did not use Jape), the variable BDEGREE3 has to be dropped from the ANOVA. It can be replaced by a variable which indicates all the degrees.

When the level of Jape usage is put into the ANOVA, none of the main effects are now significant, even though the variance explained is higher than when only the background variables were included.

Tests of Between-Subjects Effects

Dependent Variable: ItL mark

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	9997.153[a]	23	434.659	1.738	.037
Intercept	86444.372	1	86444.372	345.579	.000
BSEX	5.795	1	5.795	.023	.879
BDEGREE6	1193.376	2	596.688	2.385	.098
BPEXERS	557.913	1	557.913	2.230	.139
JUSAGEL	955.614	3	318.538	1.273	.289
BSEX * BDEGREE6	2041.547	2	1020.774	4.081	.020
BSEX * BPEXERS	640.146	1	640.146	2.559	.114
BSEX * JUSAGEL	300.181	2	150.091	.600	.551
BDEGREE6 * BPEXERS	417.946	2	208.973	.835	.437
BDEGREE6 * JUSAGEL	945.138	6	157.523	.630	.706
BPEXERS * JUSAGEL	403.672	3	134.557	.538	.658
Error	20511.791	82	250.144		
Total	397374.000	106			
Corrected Total	30508.943	105			

a. R Squared = .328 (Adjusted R Squared = .139)

BSEX: 1=male, 0=female

BDEGREE6: Degree (1=Computer Science, 2=CompSci & Maths, 3=Maths & Computing, 4=CompSci & Business, 5=Other)

JUSAGEL: Jape Usage (0=zero, 1=low, 2=medium, 3=high)

BPEXRS: Programming experience (1=some, 0=none)

Figure 130: ANOVA for background factors and Jape usage

A probable reason for the interaction between gender and degree is easy to see from the following chart:

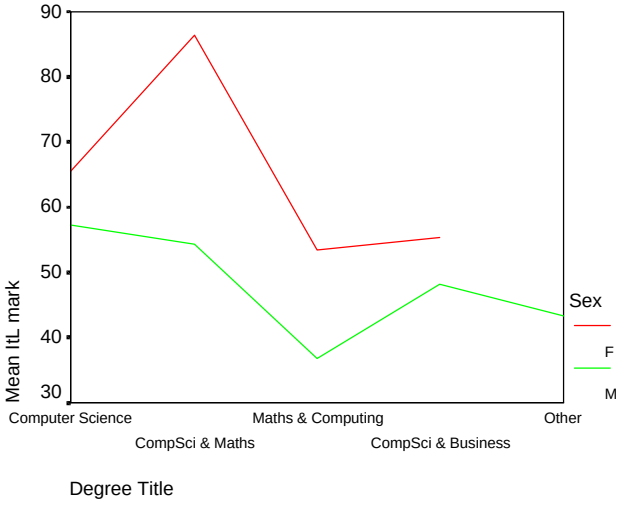


Figure 131

When the level of Jape *progress* is put into the ANOVA, it proves to be the only significant variable:

Tests of Between-Subjects Effects

Dependent Variable: ItL mark

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	12290.578[a]	22	558.663	2.545	.001
Intercept	73528.248	1	73528.248	334.983	.000
BSEX	690.812	1	690.812	3.147	.080
BDEGREE6	291.920	2	145.960	.665	.517
BPEXERS	.820	1	.820	.004	.951
JPROGREL	2163.772	3	721.257	3.286	.025
BSEX * BDEGREE6	1176.819	2	588.409	2.681	.074
BSEX * BPEXERS	249.604	1	249.604	1.137	.289
BSEX * JPROGREL	20.696	2	10.348	.047	.954
BDEGREE6 * BPEXERS	45.204	2	22.602	.103	.902
BDEGREE6 * JPROGREL	965.840	5	193.168	.880	.498
BPEXERS * JPROGREL	1445.590	3	481.863	2.195	.095
Error	18218.365	83	219.498		
Total	397374.000	106			
Corrected Total	30508.943	105			

a. R Squared = .403 (Adjusted R Squared = .245)

BSEX: 1=male, 0=female

BDEGREE6: Degree (1=Computer Science, 2=CompSci & Maths, 3=Maths & Computing, 4=CompSci & Business, 5=Other)

JPROGREL: Jape Progress (0=zero, 1=low, 2=medium, 3=high)

BPEXRS: Programming experience (1=some, 0=none)

Figure 132: ANOVA for background factors and Jape progress

A multiple regression analysis shows that prediction is not easy using such a model:

R	R Square	Adjusted R Square	SE
0.416	0.173	0.132	15.89

ANOVA	SS	df	MS	F	Sig.
Regression	5273.836	5	1054.767	4.18	0.002
Residual	25235.108	100	252.351		
Total	30508.943	105			

	B	SE	Beta	t	p
(Constant)	57.958	4.813		12.041	0.000
Sex	-8.713	4.057	-0.201	-2.148	0.034
Degree: Computer Science	-1.272	3.623	-0.033	-0.351	0.726
Some programming experience	4.644	3.262	0.136	1.423	0.158
Usage: Time spent using Jape (mins)	-0.023	0.029	-0.115	-0.796	0.428
Progress: Number of conjectures proved	0.426	0.167	0.366	2.55	0.012

Dependent Variable: ItL mark

Figure 133: Regression for overall course mark - "obvious" variables

This model suggests that the test score is significantly influenced by Jape progress and gender, but not by the degree course, any prior programming experience or time spent using Jape. However, the model explains only 17% of the variation in test score. Pragmatic choices of variables do not push this proportion much higher.

An interesting analysis is of the effect of prior attainment on Logic2 scores (here we are only considering students who sat all three tests):

Tests of Between-Subjects Effects

Dependent Variable: Logic2 test score %

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Eta Squared
Corrected Model	14617.306[a]	3	4872.435	15.277	.000	.306
Intercept	2893.555	1	2893.555	9.073	.003	.080
JPROGRE	2871.792	1	2871.792	9.004	.003	.080
OLOGIC1	5629.457	1	5629.457	17.651	.000	.145
BSEX	380.800	1	380.800	1.194	.277	.011
Error	33169.360	104	318.936			
Total	413888.000	108				
Corrected Total	47786.667	107				

a. R Squared = .306 (Adjusted R Squared = .286)

BSEX: 1=male, 0=female

OLOGIC1: Logic1 test score %

JPROGRE: Jape Progress (0-60)

Figure 134: ANOVA for Logic2, while controlling for Logic1

This table seems to suggest that performance in Logic1 can account for just 15% of the variation in performance in Logic2. Progress in Jape, meanwhile, appears to account for 8%. Gender is no longer significant as a variable, which suggests that Jape does not offer any extra advantage to females over and above that offered for Logic1.

Yet when the exam is taken, six months later, the effect of Jape progress and Logic1 performance do not appear so strong, and gender has reappeared as significant

Tests of Between-Subjects Effects

Dependent Variable: Exam score

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Eta Squared
Corrected Model	7670.207[a]	3	2556.736	8.488	.000	.197
Intercept	8370.358	1	8370.358	27.790	.000	.211
JPROGRE	1088.973	1	1088.973	3.615	.060	.034
OLOGIC1	1891.696	1	1891.696	6.280	.014	.057
BSEX	1369.189	1	1369.189	4.546	.035	.042
Error	31324.976	104	301.202			
Total	398229.688	108				
Corrected Total	38995.182	107				

a. R Squared = .197 (Adjusted R Squared = .174)

Figure 135: ANOVA for the exam, while controlling for Logic1

The best predictor of exam success is Logic2, not Logic1 or Jape progress:

Tests of Between-Subjects Effects

Dependent Variable: Exam score

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Eta Squared
Corrected Model	17014.969[a]	4	4253.742	19.933	.000	.436
Intercept	3643.360	1	3643.360	17.073	.000	.142
JPROGRE	19.099	1	19.099	.089	.765	.001
OLOGIC2	9344.762	1	9344.762	43.790	.000	.298
OLOGIC1	11.510	1	11.510	.054	.817	.001
BSEX	701.889	1	701.889	3.289	.073	.031
Error	21980.213	103	213.400			
Total	398229.688	108				
Corrected Total	38995.182	107				

a. R Squared = .436 (Adjusted R Squared = .414)

Figure 136: ANOVA for the exam, while controlling for Logic1 and Logic2

5. The Reflection Study

5.1 Data Collection

The main aim of the Reflection Study was to test the findings of the Observational Study. In-depth task-based interviews were therefore used, in which students initially tackled on paper 5-10 conjectures or partially-completed proofs from a chosen topic, and then the same conjectures were tackled using Jape. The intention was that where the paper attempts had been successful, interface issues would be highlighted when Jape was used; and where paper attempts had been invalid or had stalled, the role of ItL Jape in enabling the further development of strategies would be clarified.

The pattern of the interview was to ask for a topic that the student wished to study. Quantifiers was the most popular choice, followed by Negation. Then the first proof would be presented on paper, with questions such as "Someone was working on this proof. What's your gut reaction about what they did next?", "Why do you think they did that?", and "What would *you* do here, and why?". By this means it was hoped to find out the strategies for each rule and each global strategy, the priorities for different rules, and the behaviours when priorities fail. The conjectures used are listed in the appendix ("Prior proofs"). Students were *not* told whether they might have met this conjecture previously in the lectures, the lecture notes, Jape or the Jape manual.

Then the student would be encouraged to attempt the proof on paper, while talking through their reasoning, before moving onto the next proof. When no further useful progress was being made, the attention would switch to attempting the same proofs using ItL Jape. Students were again asked to describe their interpretation of what was going on. Differences in strategy were noted.

This plan had several advantages. It meant that students got an opportunity to remind themselves about how the interface worked while tackling proofs they could already do; it highlighted where the interface rather than students' theories might be causing them difficulties; and it highlighted where Jape assisted students.

Finally, some general questions were asked about Jape:

(i) How would you explain to someone who's never come across natural deduction before what sort of thing a natural deduction proof is, in only a few sentences?

(ii) What is it you have found most difficult about learning about natural deduction proofs?

(iii) What did you find *most useful* about using Jape to learn about natural deduction proofs?

(iv) What did you *most dislike* about Jape?

(v) What would you say was the most important thing you have got out of the logic course?

This was to attempt to ascertain perceptions of the *nature* of the activity of constructing proofs.

The 11 interviews were audio-taped (around 8 hours of paper-based work) and video-taped (about 12 hours of work both on paper and on Jape) and usually lasted around 90 minutes. 10 students were involved - 6 as individuals and 2 sets of pairs. The students were paid, but the value of the session as a means of revising for the imminent exam was also given as a possible incentive. One student (Kusi, from the Observational Study) took part in 4 sessions, and so was able to cover almost the whole range of topics.

Episode	Student(s)	Date	Topic chosen	Length of Episode (hours)
7	Kusi (S18)	15 Apr	Conjunction	1
8	Kusi (S18)	20 Apr	Disjunction	1.5
9	Kusi (S18)	20 Apr	Negation	1.5
10	Kusi (S18)	21 Apr	Quantifiers	1.5
11	Kelly (S41)	5 May	Quantifiers	2.5
12	Philip (S92)	7 May	Quantifiers	1.5
13	Charlotte (S19)	10 May	Quantifiers	1.5
14	Christian (S174)	13 May	Disjunction	1.5
15	Sita (S6) & Bibreel (S87)	14 May	Negation	2
16	Joe (S67)	14 May	Quantifiers	1.5
17	Omesh (S54) & Ibrahim (S15)	14 May	Negation	2

Figure 137: Data from Jape Reflection Study (Easter 1999)

For each episode, in addition to fieldnotes, there is a paper record of the proofs constructed by the student; an audio-recording of the student's reflections on what they are doing when attempting the paper-and-pencil proofs; a video-recording of the student's work at the computer; and sometimes a video-recording on camcorder of what the student writes during paper-and-pencil proofs.

On a methodological note, it was found that it *was* valuable seeing the video-recording of students attempting the proofs on paper prior to seeing the interactions with Jape. The paper-based work provided clues as to what students already knew and could do. It was then easy to see that certain difficulties when using Jape were related to the interface, that other difficulties when using Jape were related to the logic, and that certain difficulties when working on paper were overcome by using Jape. At times students would move back and forth between paper and Jape.

The description of episode 15 (Sita and Bibreel working on Negation) is given here as a case study in how an analysis of interactions in terms of strategies can work. This is followed by an attempt to take a broader view of the behaviours exhibited by postulating differences in prior knowledge.

5.2 Case Study Analysis - episode 15

Sita and Bibreel worked on Negation for about 2 hours. They had taken an exam for a different course in the morning, and so described themselves as being fairly tired.

Both Bibreel and Sita had A-level maths and scored slightly above average in the maths test. Neither had any programming experience. Both expected the course to be interesting and easy or fairly easy.

Bibreel was registered for "Computer Science" and motivated by a mix of factors. He scored slightly below average in Logic1 and Logic2. Until the Reflection Study he had spent just over an hour using Jape ("medium usage") but had proved all of Implication and Conjunction and 2/3 of Disjunction ("medium progress").

Sita was registered for " Computer Science & Business Studies" and was mainly motivated by following an interest. She scored slightly above average in Logic1, but collapsed to 16% (136^{th} out of 144 students) in Logic2. Until the Reflection Study she had spent just over an hour using Jape ("medium usage") but had proved only 7 conjectures (low progress), all from Implication.

The two students initially worked on proofs 18 to 22 on paper. They then tackled proofs 18 to 24 using Jape.

5.2.1 Proof 18 (PAPER) Strategies for ¬E forwards

1: P→Q, ¬¬P premises
…
2: Q

Notes

This conjecture is not in the labwork, lecture notes or Jape manual. It has been devised to test whether students have a strategy for ¬E forwards.

Response	*Suggests*
¬E on line 1.2	A strategy for recognising that ¬E forwards is useful here, perhaps akin to "Every time a double negative is seen prior to the three dots, see if removing the double negative is helpful, using ¬E forwards. "

Observations

Bibreel initially attempts to use ¬I (bad choice) rather than ¬E (good choice). Sometimes when students or experts are talking about the rules, they say "introduction" when they mean "elimination" and vice-versa (sometimes they correct themselves; sometimes others' correct them and the correction is readily accepted; sometimes the context of their speech indicates their intention). This is very often a slip of the tongue, made more likely by the constant switching between forwards and backwards reasoning (so, for example, "→I" is usually used backwards, in which case it "eliminates" the →, in some sense; but reading forwards it justifies the introduction of the →). However, in this case Bibreel did apparently mean what he said, and went down the wrong track.

Nevertheless, he and Sita eventually realise that ¬¬P ⊢ P. The simplicity of this first proof (which they had not met before) seemed to throw them.

5.2.2 Proof 19 (PAPER) Strategies for ¬I backwards

1: P→Q, ¬Q premises
…
2: ¬P

Notes

This conjecture is not in the labwork, lecture notes or Jape manual.

Response	*Suggests*
¬I on line 2	A strategy for recognising that ¬I backwards is useful here, perhaps akin to "No forward progress can be made; the ¬P could have been created by ¬I; so try ¬I backwards."

Observations

The students spot immediately that the proof would involve ¬I (good choice). They discuss how the proof will proceed after that, and then Sita says "Let's just draw the box - we're used to that" as if to indicate that drawing boxes is the way one always starts a proof.

Sita's next comment "Assume P" suggests that she may be a "forward reasoner". If in most cases students don't realise that making spurious assumptions is unnecessary if one adopts backwards reasoning, then the exemplification of this strategy in Jape (where making spurious assumptions is not allowed at all) has not triggered the realisation. There are, however, insufficient clues to be able to tell in this particular proof whether she suggested P because she noticed that assuming it would allow →E on P→Q (forward reasoning), or because assuming P was a direct result of working back from ¬P.

The proof was thereafter straightforward for the students – they had no difficulty in writing down the contradiction Q∧¬Q.

5.2.3 Proof 20 (PAPER) Strategies for ¬I backwards (after →I backwards)

Strategies for →I backwards

1:	P→Q … ¬Q→¬P	premise
2:		

Notes

This is conjecture 32 of the labwork, and conjecture 29 of the lecture notes.

Response	*Suggests*
¬I or ¬E	Rules that were previously clear are suddenly unfamiliar when negation rules become relevant.
→I on line 2	→I recognised

Observations

Bibreel's gut reaction is again ¬I (bad choice). Sita, on the other hand, prefers →I (good choice). It appears that Bibreel did notice the choice but doesn't have a reliable strategy for choosing. He explains that because ¬I is harder than →I, the former would be more likely what the person who set the exercise wanted (a second-guessing strategy). Sita, however, notes "If there is a choice between a complicated rule and an easy rule, I will go for the easy one first.". This may be sarcasm at the fact that Bibreel is ploughing ahead with his bad choice regardless of her preference, but it may be an articulation of a strategy for rule choice.

Strategies for ¬I backwards

1:	P→Q	premise
2:	¬Q … ¬P	assumption
3:		
4:	¬Q→¬P	→-I 2, 3

Response	*Suggests*
¬I applied to line 3	A strategy for recognising that ¬I backwards is useful here, perhaps akin to "No forward progress can be made; the ¬P could have been created by ¬I; so try ¬I backwards."

Observations

Having done the →I step, Bibreel isn't sure how to proceed. Does this suggest that students have internalised the →-rules on the whole, but not the ¬-rules?

When they do proceed, Sita suggests "Assume P" rather than "¬I".

5.2.4 Proof 21 (PAPER) Strategies for ¬E¬I backwards (A⊢B case)

Strategies for →I backwards

1:	¬Q→¬P … P→Q	Premise
2:		

Notes

This is conjecture 33 of the labwork, and conjecture 30 of the lecture notes.

Response	*Suggests*
→I on line 2	A strategy for recognising that →I backwards is useful here.

Observations

Sita immediately notices that Proof 21 is the previous conjecture "backwards". This suggests an awareness of structure over and above purely tactical behaviour. Bibreel says that he saw it as just another problem (i.e. that its relationship to other conjectures was not important to him).

They try →I (good choice) but give up on this strategy and tried ¬I instead (bad choice). The reasoning may have been something like "→I gave us an assumption [P] we couldn't do anything with, so let's try something else.". This may or may not be attributable to Jape encouraging them to believe that they can try something different when the strategy doesn't appear to work. Another possibility was that they reasoned that since they would have to do ¬I anyway, the →I was superfluous.

When Bibreel attempts ¬I, he carries it out correctly, having got it wrong in every previous proof. Sita notes that the contradiction (_B∧¬_B) that would be required is too complicated, and so they give up on this strategy. Bibreel comments "You can tell after a while when the method is going to work.". They return to →I.

Strategies for ¬E¬I backwards

1: ¬Q→¬P — premise
2: | P — assumption
…
3: | Q
4: P→Q — →-I 2, 3

Response	*Suggests*
¬E¬I	A strategy for recognising that ¬E¬I backwards is useful here, perhaps akin to "No forward progress can be made; so as a last resort try ¬E¬I backwards."

Observations

They now correctly apply ¬E¬I to line Q to finish the proof.

Bibreel refers at one point to a "tidy proof". Could this be a reference to a proof that one can construct using forward reasoning alone?

5.2.5 Proof 22 (PAPER) Strategies for ¬E¬I backwards (¬A⊢B and ¬A⊢¬B cases)

1: ¬(¬P∧¬Q) — Premise
…
2: P∨Q

Notes

This is conjecture 36 in the labwork and conjecture 32(b) in the lecture notes. The theorem is applied in Proof 24 below.

Response	*Suggests*
Success	A strategy for proving ¬A⊢B by turning the conjecture into ¬B⊢A: applying ¬E¬I backwards to B; the choice of A∧¬A as the contradiction; ^I to produce the line A to be proved from ¬B. This strategy may be useful especially when A is easier to break up than B. The strategy can be generalised to one that converts A⊢¬B into B⊢¬A, ¬A⊢¬B into B⊢A, and A⊢B into ¬B⊢¬A as something akin to "Find the complement and swap them round." Note that it may be used three times in this proof: once on the main conjecture, once in the subproof ¬(P∨Q)⊢¬P and once in the subproof ¬(P∨Q)⊢¬Q. Once this strategy has been developed, conjectures 37 to 42 in the labwork are straightforward.

Observations

Sita's first comment is that "There's an ∨E"; but in fact ∨E doesn't seem to apply here. Perhaps this is because she has found ∨E difficult in the past, and so it wary of the possibility; or maybe she meant "introduction".

There is a possible problem with the nomenclature "introduction" and "elimination" because of a common confusion about proof directions. This brand of logic gets its terminology from forward readings rather than forward-and-backward constructions. It might be possible therefore to teach the logic in such a way that the rules were called "∨-forwards" and "∨-backwards" instead of ∨I and ∨E. Another possibility is to change the names completely. So, for example, one could refer to "modus ponens", rather than "→E". If Latin does not appeal, one could make up names or descriptions, but this would result in students learning a logic that would appear unfamiliar to anyone else.

Bibreel's gut reaction to the conjecture is ∨E. His reasoning is «when ∨ is at the top, you do ∨E», but that he doesn't know what happens when ∨ is at the bottom. However, on second thoughts he notes that he doesn't know how to split up a negation like ¬(¬P∧¬Q). If the students were allowed to assume de Morgan's laws (or encouraged to prove them as lemmas), they may be tempted to work forwards rather than use ¬I backwards. Even better evidence for forward thinking was Bibreel's utterance during this proof: "Can I assume ¬P?". There isn't a P at the end that this could refer to, so this suggests that even when, in a previous proof, Bibreel responds to the question "Why do you want to assume P?" by referring to the P at the end, this is a post hoc explanation of why assuming P is useful rather than a description of how he decided that it was P that needed to be assumed (which might be because P→Q was one of the premises).

Sita's reaction to Bibreel's question is typical of those who are confused about the "assumption rule" described in the lecture notes for the logic course: "You can assume anything you want… Well not anything you want…". The difficulty appears to be that *on paper*, the "assumption rule" is an exploratory strategy; whereas in Jape the key exploratory strategy in connection with negation is the choice of contradiction in ¬I. The making of assumptions is not a choice for the student. The expectation is that students would experiment to find the _B in Proof 22, rather than to use an interpretation of the symbols to calculate a contradiction. The "switch-complement strategy" that allows one to turn ¬X ⊢ Y into ¬Y ⊢ X (¬E, ¬I, ∧I) has as an automatic contradiction X∧¬X. It is clear that the students are not using this strategy here, because there is no automatic selection of the contradiction.

Bibreel writes down as line 2 ¬P∧¬Q, and justifies it using ¬E1. This seems wishful thinking. It is important that students are given false conjectures once in a while - once they fail to find a proof they should explain either why any proof would fail or find a counter example or use truth tables. Although Bibreel and Sita did not face any false conjectures so far, other students seem to have been quite happy twisting the rules in order to prove the impossible.

After much effort (and several misapplied rules), the students give up on this proof, and begin to use Jape. They are encouraged by the interviewer to start with the proofs they have proved already - this will enable them to remind themselves about the interface.

5.2.6 Proof 18 (JAPE) Strategies for ¬E forwards

1: P→Q, ¬¬P premises
…
2: Q

Notes

This conjecture is not in the labwork, lecture notes or Jape manual.

Response	*Suggests*
¬E on line 1.2	A strategy for recognising that ¬E forwards is useful here, perhaps akin to "Every time a double negative is seen prior to the three dots, see if removing the double negative is helpful, using ¬E forwards. "

Observations

The students do not have great difficulty in recalling how to get the interface to do what they have already done on paper. There may perhaps be a little worry about what formula needs to be clicked. Sita clicks first on ¬¬P, and then on P→Q. Bibreel points to ¬¬P, saying "You have to click on that first, don't you.". Sita selects ¬¬P. Bibreel then points to ¬E, which Sita clicks. Sita then selects P, then Q, and asks "Which one do you click on?". She then clicks P one more time, before settling on P→Q, but the mouse pointer movements would appear rather nervous. Then she selects →E.

Bibreel, when asked how he knew that the proof is done, replies "Because there's no more gaps in the box".

Incidentally, he also refers to the "Provisos" box as being a source of slight hesitation in replying to the question, because he says isn't sure of its significance for the proof. Perhaps it would be possible to have as an option the appearance of the provisos window only when there is something in it? Or to have as another option the complete non-appearance of the provisos window, by treating provisos such as those for ∀I as *implicit*, and only referred to in an explanation window that might appear – instead of the error dialog boxes – when invalid operations are selected? This would fit with "quiet interface" principles perhaps.

5.2.7 Proof 19 (JAPE) Strategies for ¬I backwards

1: P→Q, ¬Q Premises
…
2: ¬P

Notes

This conjecture is not in the labwork, lecture notes or Jape manual.

Response	*Suggests*
¬I on line 2	A strategy for recognising that ¬I backwards is useful here, perhaps akin to "No forward progress can be made; the ¬P could have been created by ¬I; so try ¬I backwards."

Observations

P→Q is first selected. It is not clear from the video of the screen who has the mouse for this proof.

This proof is elaborated here in a little more detail than usual, because there are some interesting features about the students' inability to construct the proof, in sharp contrast to the ease with which the paper version was constructed.

Significantly, Sita says [0:04:14] "If we make an assumption… How… How are you supposed to…?". Bibreel interjects, as ¬Q is selected, "You can't write er… make an assumption. Can you. So you do that… and with… and…".

The pointer moves up and down the rules menu a few times before clicking on ∧-I. When the error message appears, he tuts and removes it from the screen within a second. P→Q is again selected. He says, "Because I don't know how you do the, um…". The pointer returns to the rules menu, before clicking on →E.

There is a 15-second pause, and then Q is selected, and then unselected, and then ¬P is selected. ¬I is selected. Bibreel says, “OK. That’s wrong”, quickly undoes the ¬I and then the →E, before re-applying ¬I to ¬P.

1:	P→Q, ¬Q	Premises
2:	P	assumption
	…	
3:	_B∧¬_B	
4:	¬P	¬-I 2, 3

He then asks how to turn _B∧¬_B into Q∧¬Q. Sita is unsure what he doing - at first (in apparent response to his question) she explains how to undo the step, and then she asks whether he wants to highlight something else.

At this point, the interviewer explains to the students how to unify (using the middle mouse button) and to pass a parameter to a rule. (Previous experience suggests that most students cannot recall either of these.)

When Sita & Bibreel use unify, initially more than just the _B is text-selected, and then just the B is selected (i.e. without the underscore character). In the past, many students have also appeared confused as to whether the underscore character is part of the unknown. Perhaps colour or a bold font could indicate an unknown, rather than using the underscore.

When Sita & Bibreel attempt to pass Q as a parameter to ¬I, they select ¬P, but then select P→Q (rather than just the Q), perhaps because the *left-hand* mouse button is used instead of the middle mouse button.

One way of bypassing the clumsiness of text-selection is to encourage students to use ∧I to break up the contradiction (so that the unknown is on a line by itself), and introduce an additional mechanism for defining the unknown (this could be in addition to the “unify” mechanism). For example, holding down for a few seconds the left-hand mouse button on the line consisting just of “_B” could make a drop-down list appear containing a range of “obvious” choices that can be selected with a single click. The final choice in the list could open up a dialog box for entering the reference for _B using the keyboard and character buttons. Alternatively for PC users, a single right-click would be a more familiar interface device than a time-delay.

Returning to Proof 19, the students apply ∧I to Q∧¬Q [0:07:41]:

1:	P→Q, ¬Q	Premises
2:	P	assumption
	…	
3:	Q	
4:	Q∧¬Q	∧-I 1.2, 3
5:	¬P	¬-I 2, 3

There is a 10-second delay and then the ∧I is undone. This seems very puzzling – why not now use →E to complete the proof?

The students now apply →E to P→Q (although there is a slight hesitation again over whether it should be P that is selected first).

1:	P→Q, ¬Q	Premises
2:	P	assumption
3:	Q	→-E 1, 2
	…	
4:	Q∧¬Q	
5:	¬P	¬-I 2, 3

After another delay, the →E is undone. Once more this seems very puzzling – why not use ∧I to complete the proof?

Although the students can carry out ∧I and →E quite easily, it doesn’t appear to occur to them that they need to carry out *both*.

After another minute of redoing and undoing these rules, Bibreel questions whether they are doing the right thing. They are asked why they think it might be wrong, and Bibreel replies “Because of these three lines.” (indicating the ellipsis with the pointer). They agree that they were expecting the three dots to disappear.

In desperation, they attempt to apply →E to P, which suggests they are still not confident about which line needs to be clicked prior to a rule application, and perhaps that they see the problem as with →E. There is more applying and undoing of →E. They even try clicking Q followed by ¬Q (as if to combine them?) but they do not attempt ∧I. Bibreel says "Don't know what to do.". Sita says "Can't see what's wrong with it.". There is some muttering as they apparently try to check the validity of the proof so far. They select P and Q∧¬Q (as if to bring them closer together and eliminate the ellipsis?). They then select ¬I, but it is not clear why. They try to apply ¬E to P.

Sita says "Shall we just leave it, or shall we just choose 'Done' and see if it works?". The interviewer hints that the →E was correct, and asks them why the dots are not disappearing. After another pause, The interviewer asks whether the proof is incomplete in some way. Sita then points to the blank justification next to Q∧¬Q and says, "Does it… Shouldn't this… Why doesn't it say the line… the proof there… that line…". The sudden incoherence suggests the vocabulary of justification appears foreign to the interactions with Jape. She then answers her own question "Because we haven't proved it, according to them.". Bibreel then points to around the Q and Q∧¬Q and says, "There should be a box here.". Sita says "Why? No.". Bibreel mumbles dejectedly "Because… Yeah, right, hmm…". Sita says "All we're doing here is just ∧I on 1 and 3.".

The interviewer suggests to them that the reason for the 3 dots might not because there is a missing line. Sita then undoes all the steps. The interviewer asks them to redo the steps and then explains that it doesn't know where line 4 (Q∧¬Q) has come from. Bibreel asks, "How can it *not know* that?" [0:13:47]. There is a sudden "Oh!" sound of insight from Bibreel, and he then selects ∧I to finish the proof.

Sita doesn't follow what was done, and asks how line 4 was completed. She does the proof again from the beginning (when prompted by the interviewer) to find out. Incidentally, when unifying _B and Q, she omits to select the underscore, and also has difficulty in selecting just the Q. Bibreel says "∧I on that…", carries out ∧I on Q∧¬Q, and then says "… and it's wrong.". He undoes it. In short, he still hasn't appreciated the nature of the difficulty. He then does →E, followed by ∧I. He quickly moves onto the next proof. There is no utterance from Sita indicating that she now understands the difficulty.

The students' odd behaviour in this episode might be explained by noting that the ellipsis might be suggesting to the students that there is a missing line (as opposed to a missing justification) - the interface might be getting in the way here - they know what the proof is but they can't finish. However, it ought to be pointed out that thanks to Jape students have until this point had no need to pay any attention to the justifications; which raises the question of when, if ever, students should have to pay attention to the justifications.

The description of this episode might seem like over-egging of a minor interface point. But for the students - who spent almost 10 minutes dealing with an essentially complete proof - it appeared far from minor.

Was this episode a waste of time for the students because the difficulty could be easily circumvented by a simple design change? (What would that design change be?) Or was it a crucial experience for Sita & Bibreel in their realisation that although the predominant metaphor for Jape is that of "simplification" there are inevitably points in proofs where simplifications do not produce new lines?

Note that whereas one might expect the *absence of a justification* on a particular line would draw students' attention to what remained to be done in the proof, this does not appear to have happened here, possibly because the automatic by-product of simplification is to provide justifications for each line, and students might have no reason therefore to pay attention to the justifications at all (let alone missing ones). Also note that *even if* this episode were an useful experience for learning that a proof can be incomplete because of a missing *justification* (rather than because of a missing *line*), judging by their second attempt at the proof, the students clearly had *not* learned this fact.

This incident does not in itself provide a criticism of the ellipsis as such, because it is a great feature that provides very useful feedback and a satisfying visual conclusion to a proof. But there is an apparent difficulty here, and it could be characterised as an interface difficulty because it does not occur on paper.

5.2.8 Proof 20 (JAPE) Strategies for ¬I backwards (after →I backwards)

Strategies for →I backwards

1: P→Q Premise
…
2: ¬Q→¬P

Notes

This is conjecture 32 of the labwork and conjecture 29 of the lecture notes.

Response	*Suggests*
¬I or ¬E	Rules that were previously clear are suddenly unfamiliar when negation rules become relevant.
→I on line 2	→I recognised

Observations

There was no hesitation in choosing →I on line 2 (good choice).

Strategies for ¬I backwards

1: P→Q premise
2: ¬Q assumption
…
3: ¬P
4: ¬Q→¬P →-I 2, 3

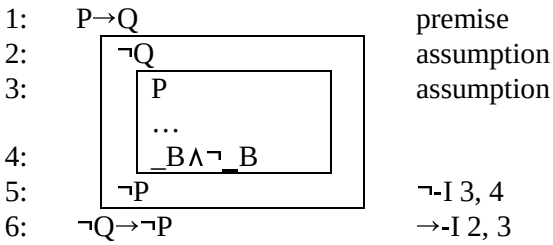
Response	*Suggests*
¬I applied to line 3	A strategy for recognising that ¬I backwards is useful here, perhaps akin to "No forward progress can be made; the ¬P could have been created by ¬I; so try ¬I backwards."

Observations

There is a short pause here. The interviewer asks what they want to do. Simultaneously, [0:17:20], Bibreel says "I want to assume P, but I don't know how you do it." and Sita says something about "Q∧¬Q" while pointing at the screen. It is not clear whether they are using the paper proof in front of them in deciding what to do next. Bibreel then says [0:17:29] "No I don't, because I know you can't do that.".

He selects ¬P, while muttering something about "Q∧¬Q", and then says "¬-introduction is it?". He selects ¬I (good choice).

Strategies for unifying variables



Response	*Suggests*
(P&P) Writes down Q∧¬Q	Realises that both Q and ¬Q can be proven.
(Jape) Unify _B and Q or ¬_B and ¬Q	Realises that _B should be Q, and can use the unify command.
→E on line 1	A strategy for recognising that →E forwards is useful here.
(Jape) Text-Select Q prior to ¬I	Realises that _B should be Q, and can give an argument to make this so.
∧I on line 4	A strategy of breaking down complex lines where possible.

Observations

Bibreel text-selects B and Q. Sita reminds him that the underscore in front of the B must also be selected ("You have to do the line in front."). She takes glee in this (because Bibreel chided her about the underscore on the previous proof), giving a little giggle. Bibreel has no difficulty in adding the underscore to the selection, and he quickly clicks "Unify".

Bibreel quickly applies ∧I to line 4 (good choice). Sita points to the Q and says "You need to prove the Q…" and points to line 1, saying "Got that from…". Bibreel has no hesitation in applying →E to line 1. The proof is complete.

5.2.9 Proof 21 (JAPE) Strategies for ¬E¬I backwards (A⊢B case)

Strategies for →I backwards

1: ¬Q→¬P premise
…
2: P→Q

Notes

This is conjecture 33 of the labwork, and conjecture 30 of the lecture notes.

Response	*Suggests*
→I on line 2	A strategy for recognising that →I backwards is useful here.

Observations

→I is selected quickly (good choice).

Strategies for ¬E¬I backwards

1: ¬Q→¬P premise
2: | P | assumption
| … |
3: | Q |
4: P→Q →-I 2, 3

Response	*Suggests*
¬E¬I	A strategy for recognising that ¬E¬I backwards is useful here, perhaps akin to "No forward progress can be made; so as a last resort try ¬E¬I backwards."

Observations

¬E is selected fairly quickly (good choice), although the mouse pointer movements are, it can be claimed, characteristic of students using the rule menu in that the whole length of the menu is traversed at least twice. Some discussion about this alleged phenomenon of inefficient mouse activity in the rules menu takes place below.

How did they know that negation was required? It is possible this is because they realise that proof by contradiction is the only possible route here; or because they have just looked down at the paper proof in front of them; or because they chose "negation" as the topic of study in the first place.

Again, when ¬I is selected (good choice), it is only after the mouse pointer has twice traversed the rules menu.

1: ¬Q→¬P premise
2: | P | assumption
3: | | ¬Q | assumption
| | … |
4: | | B∧¬B |
5: | ¬¬Q | ¬-I 3, 4
6: | Q | ¬-E 5
7: P→Q →-I 2, 6

Observations

_B and P are unified quickly (good choice) and without difficulties. Is the strategy being used to decide on the value for _B something like «Look for a likely contradiction»? In this case P is used because it has been noticed that ¬P will result from →E on lines 1 and 3. Or is it something like «Use the first simple formula.»? The former would imply a greater ability to "look ahead".

There is a short pause now, so it is not obvious that they have been planning. Sita says "Click on P∧¬P", which is then done. Almost straightaway, ∧I is selected (good choice).

1: ¬Q→¬P premise
2: | P | assumption
3: | | ¬Q | assumption
| | … |
| | ¬P |
4: | | P∧¬P |
5: | ¬¬Q | ¬-I 3, 4
6: | Q | ¬-E 5
7: P→Q →-I 2, 6

Observations

But Bibreel then says "It's not right, is it?" [0:19:17, 0580]. Sita is starting to give him instructions on what to do, saying "It's right"; but he clicks "undo". What is going on here? It is highly probable that we are seeing another example of where the ellipsis is interpreted as a missing line rather than an incomplete proof. But why is Bibreel so certain that ∧I is wrong?

Bibreel then does →E on ¬Q→¬P (good choice), followed by ∧I on P∧¬P (finishes the proof). So does he think that the order of these operations is important?

5.2.10 Proof 22 (JAPE) Strategies for ¬E¬I backwards (¬A⊢B and ¬A⊢¬B cases)

1: ¬(¬P∧¬Q) premise
…
2: P∨Q

Notes

This is conjecture 36 in the labwork and conjecture 32(b) in the lecture notes. The theorem is applied in Proof 24 below.

Response	*Suggests*
Success	A strategy for proving ¬A⊢B by turning the conjecture into ¬B⊢A: applying ¬E¬I backwards to B; the choice of A∧¬A as the contradiction; ^I to produce the line A to be proved from ¬B. This strategy may be useful especially when A is easier to break up than B. The strategy can be generalised to one that converts A⊢¬B into B⊢¬A, ¬A⊢¬B into B⊢A, and A⊢B into ¬B⊢¬A as something akin to "Find the complement and swap them round." Note that it may be used three times in this proof: once on the main conjecture, once in the subproof ¬(P∨Q)⊢¬P and once in the subproof ¬(P∨Q)⊢¬Q. Once this strategy has been developed, conjectures 37 to 42 in the labwork are straightforward.

Observations

Bibreel says "I'm actually going to see now using Jape rather than going about what we did [on paper]". [0:20:00, 0604]

On seeing the conjecture, he immediately clicks on P∨Q. The mouse pointer moves to ∨E, but then vacillates - moving up to the introduction rules, down to the elimination rules, up again to the introduction rules, down again to the elimination rules, and finally clicks on ¬I (bad choice).

The error message that ensues is immediately dismissed. Presumably, then, Bibreel has no strategy that says something like «¬I can only be applied backwards to a formula with the principle operator ¬.»; but he *does* have a strategy that says «Treat error messages merely as signals that this operation is inapplicable. Reading them is a waste of time.».

He clicks on ¬(¬P∧¬Q), and this maintains the selection of P∨Q, so he clicks outside the proof and appears to be about to select ¬(¬P∧¬Q) alone. However, he instead selects P∨Q again.

The mouse pointer goes to the elimination rules, and then up to the introduction rules, as Bibreel says "Where is it?". ∨I(L) is then selected (bad choice).

At this point one would expect a student with a strategy along the lines of «Check if new lines produced backwards can be proved from earlier lines.» would undo, because proving something simple (i.e. P) from something complicated (i.e. ¬(¬P∧¬Q) on line 1) would appear to be worth checking, either by creating an everyday model for the symbols, or by examining truth values.

Instead, ¬E¬I is applied to P. Is simplicity the reason why ¬E¬I is the "obvious" operation for P but not for P∨Q?

The new unknown ¬_B is unified with ¬(¬P∧¬Q). There is a pause; this step is undone; it is redone.

∧I is applied to line 4 (good choice).

1:	¬(¬P∧¬Q)	premise
2:	¬P	assumption
	…	
3:	(¬P∧¬Q)	
4:	(¬P∧¬Q) ∧ ¬(¬P∧¬Q)	∧-I 1, 3
5:	¬¬P	¬-I
6:	P	¬-E 5
7:	P∨Q	∨-I(L) 6

Jape leaves the brackets around ¬P∧¬Q because it tries to work with the user's entries as much as possible rather than with simplifications. However, in this case the brackets are perhaps not helpful. Nevertheless, Bibreel is apparently not thrown by them, because he has no hesitation in applying ∧I to line 3 (good choice).

Sita asks "Where did you get that from?". It is difficult to make out Bibreel's reply here. Sita says "We assumed ¬P so you can assume ¬P∧¬Q." This doesn't seem to make much sense, a fact she appears to acknowledge when she says "Hang on…".

There is a long pause, and then Bibreel says "I'll tell you what to do" and applies ¬I to ¬Q. This is undone almost straightaway. He says, nevertheless, "I've kind of seen it.". Sita asks him, "What does it look like?". He starts to say something, and then falls silent.

Sita asks to see again what Bibreel did before, and so he redoes the step ¬I applied to ¬Q.

There is a long pause now, while they undo and redo some of these steps.

At one point Bibreel says "Hang on, but what's the problem?" and attempts to apply ∧I to ¬Q. Shortly afterwards he tries to apply ¬E to ¬(¬P∧¬Q).

Sita says "You have to *assume* something else as well." [0:25:54, 0728], which suggests that her fallback strategy is «If you get stuck, assume something.».

∧E(L) is applied to ¬(¬P∧¬Q), which of course fails to work. One wonders how secure knowledge of ∧E can be if, admittedly in desperation, this step is attempted.

The interviewer draws Sita and Bibreel's attention to the question of whether line 3 can in fact be proved. Bibreel's initial response is to look for a menu item or button that would tell them that. Sita's initial response is "We need something... [as an assumption?]", whereas the intention was to raise doubts about whether an incorrect step had been taken. Even after the interviewer makes a heavy hint that they have gone wrong, Sita says "If we make another assumption...". What this might suggest is not only that she believes the way out of any impasse is to make an assumption, but also that while it is possible to make an *unnecessary* step it is not possible to make an *incorrect* step that renders the proof impossible.

Bibreel says that he thinks they have gone wrong, but when asked how he knows, he has no answer (in other words, he is just inferring it from the hints). He also says "You know what it is - it's just that I don't want to give up this much" as he indicates the expanse of the proof so far. Sita laughs and adds "After all this time.". They consider starting the proof again. Before they do so, they are again asked how they might be able to tell whether they have gone wrong. Bibreel does not have an answer, other than that the first ¬ in line 1 is "causing all the problems. The only way you can get rid of a 'not' is by ¬... er... elimination.". He then tries applying ¬E to ¬(¬P∧¬Q) once again.

The interviewer suggests to them that they look at the steps they have taken to see where they had choices. They now start to undo steps [0:28:23, 0786] right back to the start.

Bibreel clicks on P∨Q and says "That *has* to be ∨ introduction." (bad choice). Why doesn't he see that ¬E¬I is another possibility? They have already used it in this proof, and had no apparent difficulties with it in Proof 21 (either on paper or in Jape). One explanation could be that on those previous occasions it was applied to a single proposition ("P", for example, or "Q") rather than a formula ("P∨Q", for example, or "P→Q"). Samin (lab assistant) portrays the situation as one of understanding that P≅¬¬P so that one can always replace P by ¬¬P, and ¬¬P by P.

The pointer lingers on the elimination rules before finding ∨I(L) and ∨I(R). Sita asks "What's the difference between those two?". Here, then is an opportunity to realise that the fact that a choice is necessary between left and right may be an indication of error. However, Bibreel just replies "Because you could have P to prove, or Q". Sita says "OK." and Bibreel selects ∨I(L) with no further discussion (bad choice). He says "Thank you" (to the computer, in appreciation that it has done what he expected it to do?), and then asks, "Now, so if you've got P, how

are you going to prove P?". Again, here is another opportunity to check the possibility of proving a new line from the premise. Sita says something that cannot be heard. Bibreel asks "Can we prove P?".

Perhaps it is an unreasonable expectation that students will use a checking strategy. After all, does it feature in the lecture notes? It could be seen, moreover, as an alternative to the use of natural deduction rules, and therefore as an invalid (or unnecessary) procedure.

Bibreel goes on, "You see, straightaway I've got to do the complement [which is presumably his name for ¬E]. How will I prove P... other than to do...?" and he clicks ¬I (P is selected), which brings up an error message. He then clicks outside the proof (de-selecting P) and click ¬E. "Other than to do that, I don't know what to do.".

The interviewer then draws their attention to what they did on paper. They immediately undo all steps and apply ¬E¬I to P∨Q (good choice). They had attempted something like this on paper (although it was incorrectly followed through, and they abandoned the approach because of the complexity).

1:	¬(¬P∧¬Q)	premise
2:	¬(P∨Q)	assumption
	…	
3:	_B∧_¬B	
4:	¬¬(P∨Q)	¬-I 2, 3
5:	P∨Q	¬-E 4

Bibreel text-selects ¬_B and ¬(¬P∧¬Q), saying "That [¬_B] should be... that [¬(¬P∧¬Q)]" (good choice). When asked how he decided on this unification, Bibreel replies that "It has to be a contradiction using a premise you already have.".

Unfortunately, the question may have inadvertently partly dissuaded Bibreel from his plan. He attempts to apply ∨E to ¬(P∨Q) (bad choice), saying "Can you do this?". When this fails, he returns to the unification. Sita appears to be attempting to disagree with the plan, but Bibreel goes ahead anyway. Incidentally, Jape shows ¬P∧¬Q ∧ ¬(¬P∧¬Q) instead of the more natural (¬P∧¬Q) ∧ ¬(¬P∧¬Q).

He then applies ∨I(L) to ¬P∧¬Q ∧ ¬(¬P∧¬Q) (peculiar choice). There is no indication whether this was a slip of the mouse, but it seems likely. When this fails, he applies ∧I the same formula (good choice).

1:	¬(¬P∧¬Q)	premise
2:	¬(P∨Q)	assumption
	…	
3:	¬P∧¬Q	
4:	¬P∧¬Q ∧ ¬(¬P∧¬Q)	∧-I 1, 3
5:	¬¬(P∨Q)	¬-I 2, 4
6:	P∨Q	¬-E 5

He says "Still not going very well.". Nevertheless, he has now effectively removed - although he does not appear to have noticed it perhaps - that troublesome negation from line 1 that he discussed before. He can now split up the ¬P∧¬Q and prove each of them from ¬(P∨Q).

He attempts to apply ∨I(L) to ¬P∧¬Q (another odd choice - another mouse slip? He says "I've done it again"), and then ∧I to the same formula (good choice).

1:	¬(¬P∧¬Q)	premise
2:	¬(P∨Q)	assumption
	…	
3:	¬P	
	…	
4:	¬Q	
5:	¬P∧¬Q	∧-I 3, 4
6:	¬P∧¬Q ∧ ¬(¬P∧¬Q)	∧-I 1, 5
7:	¬¬(P∨Q)	¬-I 2, 6
8:	P∨Q	¬-E 7

Sita asks "What happened there?". Perhaps this indicates that she might be intimidated by the bifurcation.

Bibreel's reply to Sita is not audible. What he does next is to point to ¬(P∨Q) on line 2, and say that the negation is now the problem because, "You can't do anything with the P or Q [or P∨Q, perhaps] now". There is a long pause, and then they agree that this approach is not getting anywhere.

Were students using Jape's greying-out to help them decide where in the proof to pay attention.

They start to undo all the steps, but time is running out and so the interviewer tells them that they were on the right lines. They redo all the steps. There is another long pause. The interviewer points out that if they can't split up line 1 and they can't split up line 2 then they can't move *forwards*. Bibreel takes the hint and applies ¬I to ¬P (good choice), saying "You're joking, I know it's not this.".

1:	¬(¬P∧¬Q)	premise
2:	¬(P∨Q)	assumption
3:	P	assumption
	…	
4:	_B∧¬_B	
5:	¬P	¬-I 3, 4
	…	
6:	¬Q	
7:	¬P∧¬Q	∧-I 5, 6
8:	¬P∧¬Q ∧ ¬(¬P∧¬Q)	∧-I 1, 7
9:	¬¬(P∨Q)	¬-I 2, 8
10:	P∨Q	¬-E 9

Bibreel guesses that the contradiction has B = P∨Q (good choice), and tries it, saying, "I'll just take a look.". There is a short pause than he applies ∧E(L) (bad choice). He realises from the output that that was the wrong rule, and applies ∧E(R) instead (just as bad). Sita is giggling, presumably at the complexity of the proof. Then he realises that it should have been ∧I (good choice). He blames tiredness for the error.

He correctly applies ∨I(L) to P∨Q fairly quickly (good choice), and says, "And then the same thing with Q.". He says to Sita, "You do it. I'm tired.", which could be interpreted as "I've done it, but I don't know how.".

Sita says she wants to look through the proof so far. Bibreel says he couldn't believe that it was going to be so complicated: "It can't possibly want me to do *all* this.".

The interviewer points out that there were three ¬I's.

Bibreel says, "Because I was following standard procedures and rules, and I knew that I couldn't go using the ¬'s. [inaudible] Then it got even more complicated.". Sita meanwhile has applied ¬I to ¬Q and unified _B with P∨Q. She selects Q, but Bibreel says "No. ∧I. Line 8". She select line 8 - (P∨Q) ∧ ¬(P∨Q) - and then ∧E(L). Bibreel says "Ah no", and Sita clicks on "Redo" (by mistake) at the same time as Bibreel answers the question about why it should be ∧I with "Because it has been AND-ed there" while pointing to line 8. However, redoing the step causes the proof to move up the screen slightly, just as his hand is blocking the view. Sita clicks on "Redo" again while the interviewer asks "What happened there?". She says "I don't know." Bibreel has spotted that she was clicking "Redo" instead of "Undo", and clicks on "Undo" twice to compensate. The top lines of the proof now disappear off the top of the screen. Bibreel asks "Where's it gone?". Everyone is flustered by this.

He clicks "Undo" once more, and then applies ∧I to line 8, as was intended in the first place. The interviewer, suspecting that Bibreel has discovered that ∧I is useful for splitting up the inevitable ∧-line that ¬I produces, asks him why he always selects ∧I. He replies, "Because of that ∧ [indicating with the pointer the central ∧ in (P∨Q)∧¬(P∨Q)]… In order to get from there [(P∨Q)] to there [(P∨Q)∧¬(P∨Q)] you have to have the ∧." [0:36:45, 0947]. This suggests that he is still working on a rule-by-rule basis and does not relate ∧I to a ¬I∧I strategy. It does seem plausible, though, that he is reasoning backwards, but his explanation is forwards («∧I is right because you need an ∧ to get from P∨Q to (P∨Q)∧¬(P∨Q)») rather than backwards («(P∨Q)∧¬(P∨Q) must have arisen by ∧I»).

The proof is complete when Bibreel applies ∨I to P∨Q.

Sita twice asks "Why did they put that in brackets?", indicating the line "(P∨Q)".

In summary, then this proof was too difficult for these students on paper, and only achievable with Jape because of heavy hints.

5.2.11 Proof 23 (JAPE) Strategies for ¬E¬I backwards (⊢B case)

Strategies for ¬E¬I backwards

…
1: P∨¬P

Notes

This is conjecture 34 in the labwork and conjecture 31 in the lecture notes. It is also used as an example in the Jape manual. The theorem is applied in Proof 25.

Response	*Suggests*
¬E¬I	¬E¬I has been developed as a unified strategy
¬E	Strategy for recognising that ¬E may be useful here.

Observations

The interviewer suggests that this proof is hard and that it might be a good idea to tackle it using Jape at first, rather than using pencil-and-paper. He says that it is in the notes, and Sita says that she remembers it from the notes.

The pointer moves to ¬E [0:38:06], then to ¬I, and then back to ¬E, which is clicked. ¬I is applied to ¬¬(P∨¬P) almost straightaway. (It is not clear from the videotape who is controlling the mouse). This suggests a unified ¬E¬I strategy that can be applied as well to a disjunction as to a single proposition, in contrast with the previous proof.

Strategies for unifying variables after ¬I backwards

1:	¬(P∨¬P) … _B∧_¬B	assumption
2:		
3:	¬¬(P∨¬P)	¬-I 1, 2
4:	P∨¬P	¬-E 3

Response	*Suggests*
(P&P) Write down (P∨¬P)∧¬(P∨¬P)	Realises that the contradiction can be little else.
(Jape) Text-Selection of P∨¬P before ¬I	Can give an argument to ¬I using Text-Selection.
(Jape) No text-selection	Ask if there was any way the contradiction could have been defined prior to ¬I
(Jape) Unify _B and P∨¬P	Realises that there is little else _B could be, and can use the unify command.

Observations

First, ¬_B is unified with ¬(P∨¬P).

Then, almost straightaway, ∧I is applied to the contradiction [0:38:27, 0997]. This suggests, again in contrast with the previous proof, that ∧I is part of a strategy «¬E - ¬I - unify - ∧I».

The question of what formula should be unified with the unknown remains unclear because there is little choice in this case. The interviewer asks about the reasons for the choices here, and Bibreel replies:

Bibreel: You know with things like this - say with maths - you go on a lot of *gut feeling*. You go on a lot of *instinct*, rather than… thinking like…When you started asking, I started thinking, 'Oh my God…'

Sita: … *why* am I doing this?

Bibreel: And it's just you go from one step to another to another. And things are just making you go in that direction. You're just flowing. And you don't necessarily *know* why. There's a lot of gut feeling; there's a lot of instinct… which I've learned, actually…

Shortly after, he goes on:

Bibreel: … Definitely when you are doing things like this - and maths - generally procedures - procedure questions - when you've done it enough times then it's all about instinct and what you think…erm… But yeah, if you want to ask… now I'll probably look back and think 'Yeah, why *did* I do that?'!

1:	¬(P∨¬P)	assumption
	…	
2:	(P∨¬P)	
3:	(P∨¬P)∧¬(P∨¬P)	∧-I 2, 1
4:	¬¬(P∨¬P)	¬-I 1, 3
5:	P∨¬P	¬-E 4

This is perhaps the trickiest part of the proof because it may look like no progress has been made. But of course there is now an assumption that can be used. The students' attempts in the previous proof would suggest that they may try ∨I on line 2 (good choice), but *not* because they have checked that P could be plausibly proven from ¬(P∨¬P); rather because they appear to have a strategy along the lines of «Always try other possible rules before resorting to the negation rules».

Action (Bibreel, 0:40:25): ∨I(L)

This step was as predicted.

1:	¬(P∨¬P)	assumption
	…	
2:	P	
3:	(P∨¬P)	
4:	(P∨¬P)∧¬(P∨¬P)	∧-I 3, 1
5:	¬¬(P∨¬P)	¬-I 1, 4
6:	P∨¬P	¬-E 5

In terms of efficiency, ∨I(R) might be better than ∨I(L) because the latter requires ¬E before ¬I; but this is a minor point.

A couple of seconds passes and then:

Bibreel: No no no.

{ **Interviewer:** Why do you say that?

Sita: What's wrong with that?

Action (Bibreel, 0:40:31): Mouse pointer hovers over "Undo".

It is possible that a primitive strategy of checking for plausibility by informally interpreting the symbols could run into difficulties with the proof at this point. Can P be proven from ¬(P∨¬P)? Normally one can conclude that if neither A nor B is true then they both must be false (an informal version of one of De Morgan's laws). So if neither P nor ¬P is true then they both must be false. That is: P is false and P is true; which is fine, because that is the whole point of this reductio, and so P can be proven. However just looking at the first part of the informal conjunction might lead one to the idea that P has to be proven from P being false, which isn't plausible.

However, Bibreel's subsequent words and actions suggest that he isn't paying attention to semantic considerations anyway.

Bibreel: Because I want to do the same thing. [meaning ¬E¬I perhaps?] Would you have done that?

Sita: Yeah.

Action (Bibreel, 0:40:37): Clicks "Undo".

Bibreel: Because I want to do this...

Action (Bibreel, 0:40:40): Selects P∨¬P then ∨I(R).

Bibreel: It shouldn't make a difference either way to be honest.

This suggests that having done ∨I(L) he realises that he will have to apply ¬E before ¬I, but that he could just as easily have produced ¬P rather than P, and in that case ¬E would not be necessary. He also notices that there is not much difference between the two methods in any case.

He then appears to follow the «¬E¬I∧I» strategy: he applies ¬I to ¬P, unifies ¬_B with ¬(P∨¬P) and applies ∧I to the contradiction, all without much hesitation. After a little bit of mouse movement he applies ∨I(L) and so finishes the proof (0:41:15).

What we don't know is whether Bibreel spotted that the proof would reduce to proving P∨¬P from P (or ¬P) when he applied that first ∨I step at 0:40:25. «¬E¬I∧I» can work just as a handy heuristic, or it can be used with the suspicion that it should simplify things, or it may be used with an accurate prediction of exactly *what* will result.

Bibreel says (with feeling) that his pride has been restored - he had been very annoyed at struggling with the last proof. This one was "not necessarily easier than the last one".

Sita says, when asked if she has followed what Bibreel has done, that she thinks she would have been able to do it.

5.2.12 Proof 24 (JAPE) Sequent Introduction using ¬(¬P∧¬Q) ⊢ P∨Q

Strategies for →I backwards

…
1: ¬(¬P∧¬Q) → P∨Q

Notes

This conjecture is not in the labwork, lecture notes or Jape manual. The theorem ¬(¬P∧¬Q) ⊢ P∨Q has been proved in Proof 22 above.

Response	*Suggests*
→I on line 1	A strategy for recognising that →I backwards is useful here.

Observations

Sita (0:42:51): I don't like these ones... that start without... [premises]

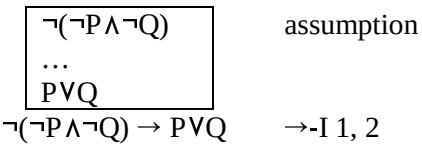
Bibreel: I know what to do... Implies-introduction.

Action (Sita): Selects line 1. Initially points at →E.

Bibreel: No.

Action (Sita): Clicks on →I.

Strategies for Sequent Introduction



Response	*Suggests*
Applies ¬(¬P∧¬Q) ⊢ P∨Q	A strategy for recognising that SI is useful here.
Difficulties	Ask "Could you use another proof that you might have already completed to help here?"

Observations

Bibreel: Or-introduction.

Action (Sita): Clicks P∨Q. Mouse pointer moves hesitantly up and down the rules menu, before clicking ∨I(R) (bad choice).

Before this step the proof was effectively Proof 22 - the proof that caused so much angst earlier. There is no evidence that the students notice this (such as commenting on the similarity or attempting Sequent Introduction). Moreover, they do not appear to have learned as a result of Proof 22 the need for a checking strategy. Once again the preference is for ∨I over ¬E¬I. But without a semantic check is it possible to realise that this avenue is a dead-end? Just as seriously, it appears from what follows that the students do not realise that an alternative choice is possible at this step. They either need some sort of background mental flag to be triggered here that a major proof-choice has been made where another is possible; or a strategy - when stuck further in the proof - of undoing actions step-by-step, checking if there were alternative choices.

Action (Sita, 0:43:28): Selects Q.

Bibreel: Um, not... elimination.

Action (Sita, 0:43:32): Selects ¬E. Selects ¬¬Q.

Bibreel: Not introduction.

Action (Sita, 0:43:39): Selects ¬I. Selects the line _B∧¬_B.

Bibreel: Um...

Action (Sita, 0:43:45): Clicks outside the box. Starts to text-select ¬_B.

Bibreel: (pointing at ¬_B) That's the same as that (pointing at ¬(¬P∧¬Q)).

Action (Sita, 0:43:51): As she attempts to text-select ¬(¬P∧¬Q), the whole proof moves slightly to the right. Says something inaudible. Successfully unifies the two formulas.

Bibreel: And you should...

Action (Sita, 0:44:00): Selects the contradiction, and hesitates.

Bibreel: No, no! Oh yeah. And and-introduction (points to ∧I).

Action (Sita, 0:44:09): Selects ∧I.

1:	¬(¬P∧¬Q)	assumption
2:	¬Q	assumption
	...	
3:	¬P∧¬Q	
4:	¬P∧¬Q∧ ¬(¬P∧¬Q)	∧-I 3, 1
5:	¬¬Q	¬-I 2, 4
6:	Q	¬-E 5
7:	P∨Q	∨-I(R) 6
8:	¬(¬P∧¬Q) → P∨Q	→-I 1, 7

Sita seems to know at every step which expression should have the focus - she usually selects it before Bibreel says what rule to apply.

Action (Sita, 0:44:12): Selects line 3 (¬P∧¬Q)

Bibreel: And-introduction here.

Action (Sita, 0:44:14): Selects ∧I.

How Sita would fare if Bibreel weren't dominating the action is difficult to say.

Action (Sita, 0:44:19): Selects line 3 (¬P).

Bibreel: Um... you know what? Oh yes I see...

Sita: And.

Bibreel: Not.

Sita: (mouse lingers over ∧I) And.

Bibreel: No. (inaudible) Not. (inaudible)... introduction? See what not-introduction does.

Action (Sita, 0:44:25): Selects ¬I.

Bibreel then tells Sita that ¬_B is ¬Q, so she unifies the two. It looks as though Bibreel might be using a strategy such as «To choose the contradiction, use the first assumption - reading downwards - that has not already been used in a contradiction.».

Bibreel now has little hesitation in recommending ∧I after unification. It is possible (although unlikely, given his responses) that this improvement to the ¬I strategy is because the interviewer drew attention to ∧I by his questions. More likely is that Bibreel has realised using Jape that this is always a clarifying action after ¬I.

Bibreel (0:45:03): And-introduction. (pause) And you know what - I still don't know whether it's right or wrong. But you just hope somewhere along the line it's going to fall into place...

Action (Sita, 0:45:13): Selects Q∧¬Q and then ∧I.

1:	¬(¬P∧¬Q)	assumption
2:	¬Q	assumption
3:	P	assumption
	...	
4:	Q	
5:	Q∧¬Q	∧-I 4, 2
6:	¬P	¬-I 3-5
7:	¬P∧¬Q	∧-I 2, 6
8:	¬P∧¬Q∧ ¬(¬P∧¬Q)	∧-I 7, 1
9:	¬¬Q	¬-I 2-8
10:	Q	¬-E 9
11:	P∨Q	∨-I(R) 10
12:	¬(¬P∧¬Q) → P∨Q	→-I 1, 11

Bibreel (just after ∧I is applied): ... And it should have done - just then. ... I think. (pause) I don't know what's going on. It's doing one of those things where it's not... where the computer's misbehaving. Because everything's there, and it's just not....

At this stage, a checking strategy would be helpful to forestall further exploration down this avenue. However, it is not screamingly obvious that things have gone wrong (but it should be possible for the students to discover that Q cannot be proved from this set of assumptions).

Action (Sita, 0:45:30): Selects Q and then ∧I.

Sita: Hang on a minute...

Bibreel: Because it's got the Q and the ¬Q. I don't know what its problem is.

Bibreel appears to believe that the Q is proven. He seems to think that this is similar to the "problem" they met earlier when the ellipsis indicated a missing justification rather than a missing line. However, he appears to have mistakenly thought that Q is proven. Sita appears to realise his error, but doesn't have the confidence to say so:

Sita: How did we get the... (indicates the line "Q" with the mouse)... Q.

Bibreel: Ah (in exasperated resignation), it's one of those things...

Sita: Hang on, we've already got Q... (indicating line 11)

Bibreel (0:45:52): We've Q and ¬Q so everything's proven. It just that there's something... It wants you to tell it what to do. So... what, I mean...?

This is bizarre. Not only does Bibreel apparently believe - all of a sudden - that the mere display of a formula on the screen (or its generation from other formulae) means that it is proven, but Sita appears to have the theory that later lines can act as justifications for earlier lines, thus undermining the whole activity. This joint mental aberration could be blamed on the fact that the students have been concentrating hard for well over an hour and are tiring. It certainly does not fit with the view that this illustrates that the students have learned superficial proof mechanisms but have not developed "true understanding". There is nothing new here that challenges the students to reflect on their understanding.

Action (Sita, 0:46:05): Selects Q and ∧I.

Is Sita trying to "apply" the justification shown on line 5, here? It seems improbable that she believes that ∧I can apply to Q.

Action (Sita, 0:46:10): (muttering something) Selects ¬Q (Q remains selected also) and then ∧I.

Bibreel: What are you doing? (pause of 20 seconds) It is proved isn't it?

Interviewer: Um...

Bibreel: Because all the elements are there (points at the proof).

Bibreel (0:46:58): Oh! (insight).

Sita: I don't think...

Bibreel: It's saying 'Where are you going to get the *Q* from?'.

Sita: Yeah...

Bibreel: Maybe...

Sita: Yeah, that's what it saying - 'Where did you get the Q from?'.

Action: There is a pause of 20 seconds, and then somebody starts undoing steps. Three steps are undone, and then Sita says something about a number while pointing at one of the justifications. The third step is then re-done, but this could be just as easily to do with the disappearance of the innermost ¬I box as with interpreting Sita's comment as the start of an objection to the third undo.

Providing a button for undo would enable those who wanted to compare "before-and-after" to avoid having the proof partially obscured by the pop-down menu. The 'Backspace' key can be used at the moment, but none of the students appear to know this.

Action: The mouse pointer lingers over the contradiction _B∧¬_B before two more steps are undone. ∧I is applied to ¬P∧¬Q (to check if it was done correctly?). This is undone, and one more step is undone. Sita starts to say things

on several occasions. ∧I is applied to ¬P∧¬Q∧¬(¬P∧¬Q), then ∧E(R) is applied to ¬P∧¬Q. This is undone, and ∧I is applied to ¬P∧¬Q. ¬I is applied to ¬P.

What appears to be happening is that the proof so far is deconstructed a few steps, each step is then checked (to see if is plausible, or if there were alternatives?) and then a few more steps are undone.

Action: This time, ¬_B is unified with ¬(¬P∧¬Q) rather than with ¬Q. ∧I is applied to the contradiction, followed by ∧I to ¬P∧¬Q.

In principle, this could go on for ever, while ¬I is applied with different contradictions to different parts of the proof. In practice the students do not feel happy about exploring the proof to a length much more than 12 lines. They could be left to their own devices, in which case the exploration could be continued, or they might give up, or they might eventually decide to restart, or they might undo as far back as that erroneous second step. A teacher who knew they had gone wrong might suggest one of the latter two courses.

Sita: That's not right is it?

Bibreel: No.

Action: The three most recent steps are undone.

Bibreel: We were better off the last time [when _B was Q rather than ¬P∧¬Q, presumably]

Action: ¬_B is once again unified with Q. ∧I is applied to Q∧¬Q.

Sita: It's what we had last time, isn't it?

Bibreel: Except I don't know where we get the Q from.

Action: The mouse pointer indicates the Q on line 4. Then it circles the Q on line 10, before moving up to line 4 again, then down to line 10 again.

Sita: You have to do...

Bibreel: Hang on. Hang on. Can't do that... because look...

Sita: ... we've got Q down there.

Bibreel: Yeah, it's exactly the same situation as we've got down there, anyway.

Action: Redo is selected. This may be a misclick for 'undo' because it has the unexpected effect of reverting to the ¬P∧¬Q substitution for _B. Redo is clicked five more times, before Bibreel says "Whoops" and clicks undo three times. There is a pause of 10 seconds, followed by another undo, followed by another pause. Then there is a redo, then two undoes. There are another four undoes, and then a pause (when the innermost ¬I has disappeared).

The interviewer (0:52:07): Now we've actually been in this situation before... a similar situation.

Sita: Yeah, I was going to say. It's similar to that other question.

Bibreel: That's good because it means there's a pattern, a certain thing that I keep falling for. And I want to find out what it is.

I think "it" is a default preference for ∨I over ¬I.

The interviewer then asks them if they have a way of telling whether they can prove line 3 (¬P). Line 1 says ¬(¬P∧¬Q) and line 2 says ¬Q. There is 30 second interruption at this point, during which time somebody undoes the step that produces line 3 (∧I applied to ¬P∧¬Q). They have no reply, so the interviewer suggests that an informal way of interpreting line 1 is that it excludes ¬P and ¬Q being both provable.

Sita: Because you forget about what it actually means...

The interviewer gives them a debriefing on this proof. He asks them to undo a few more steps until:

1:	¬(¬P∧¬Q)	assumption
	…	
2:	Q	
3:	P∨Q	∨-I(R) 2
4:	¬(¬P∧¬Q) → P∨Q	→-I 1, 3

He then suggests that it is not possible to prove line 2 from line 1. There is no indication from the students that they see that. He asks what else they could do other than ∨I.

Bibreel: Ah! (insight) You negate that!

They decide to finish off the proof (0:57:30), but this is not so straightforward.

Action: ¬I applied to P∨Q (can't be done). ¬E; ¬I; unification of _B with ¬P∧¬Q; ∧I; undo; undo; unification of _B with P∨Q; ∧I; ¬E; ¬I; undo; undo; ∨I(L); ¬E; ¬I; unification of _B with P∨Q; ∧I; undo 5 times; ¬E; ¬I.

The interviewer then looks at Proof 22 and realises that when he wasn't looking Bibreel changed the contradiction for the first ¬I from (¬P∧¬Q)∧¬(¬P∧¬Q) to (P∨Q)∧¬(P∨Q). This is corrected, and the rest of the proof follows fairly easily without further hints.

5.2.13 Interview Questions

Interviewer: How would you summarise what it is you found most difficult about learning natural deduction?

Bibreel: About natural deduction?

Interviewer: Yeah, learning to prove. ... What are the things you found most difficult?

Sita: I found... just, um, the most difficult was or-elimination. I can't even remember that now. But that doesn't come up that much, really. Does it?

Bibreel: I don't think there's a specific, um, *rule* or anything... that, you know, you can say 'That was particularly hard'. But they can tease you... how they put together questions. And... little tricks and things. I mean, you see, for me, there was a consistent pattern where I kept falling through the trap, or whatever. So if I took care of that...

Interviewer: (*to Sita*) But for you... or-elimination?

Sita: Yeah, and I just have to, basically, you know, try and remember the rules... That negation one... takes me a bit longer to... work out what's going on. But I think that with a bit more revision I should be alright.

Interviewer: And what would you say you found most useful about using Jape to help you?

Bibreel: It's quick.

Interviewer: It's quick.

Bibreel: It is quick. And you can trial-and-error a lot. I trial-and-error anyway, but if you saw how I answered even some of the earlier questions, I trial-and-error. I do it, and then if it doesn't work out I'm onto the...

Sita: But in an exam, you try and do it without...

Bibreel: I've learned to do that quick enough that I can do it in the exam as well. But in the exam, they tend not to make it so that it's going to take you that long. But with Jape you can do questions like this and I'm basically trial-and-error again but you can do it so much quicker.

Interviewer: (*to Sita*) Is it the same for you?

Sita: Yeah, and it tells you... I mean if you do it all wrong on paper, I mean you realise after a few minutes that you've done it wrong but on this it'll tell you straightaway.

Interviewer: What did you most dislike about Jape?

Bibreel: Um... (*long pause*) I think the only thing that got a little bit annoying was the... there was only one question where everything was there and you're thinking, you know, "The computer should be able to take it from there.". But they wanted you to do every single detail. But I think that's just computers for you, isn't it. (*pause*) Not that there not efficient, but they want you to do...

Interviewer: Yes.

Sita: Same. Nothing really.

Interviewer: What would you say was the most important thing that you've got out of the logic course as a whole.

Sita: (*pause*) You know the beginning... Not the beginning, sorry. The 'all' and 'some', the way they write it out with this (*inaudible*) all frogs are green or that... I guess that makes you think about things a little bit more.

Bibreel: Sounds silly, but, yeah, I was going to say - it makes you think logically! (*Sita laughs*) Which is why I guess it's called 'logic'!

Interviewer: Because some people would say "Oh, it's just a game you play..."

Sita: It's more fun than...

Bibreel: You can see it like that. If you think of it in terms of that you'll enjoy it more, doing it more, if you think of it in terms of a game. 'Cause you'll want to do these questions more. But then if you look at the questions and see how it's benefiting you, it is... very logical.

Interviewer: And final question: How useful would you say today's session has been for you?

Sita: Quite useful.

Bibreel: For me too.

5.3 Further observations from the Reflection Study

Many of the findings from the Observational Study were corroborated by students' actions and responses in the Reflection Study. For example, with the exception of the forward reasoners, students generally found it straightforward to use the program to explore proofs. The text of dialogue boxes was ignored. Difficulties were found with interpreting Jape's execution of incomplete steps, with appreciating whether it is ∧E(L) or ∧E(R) that concludes P from P∧Q, with carrying out text-selection, with understanding that it is not possible to create P∧Q by ∧I on P and Q, and with handling the variables used in Quantifier conjectures. The confusion, reported in the Observational Study, about the direction in which a rule would be applied was reproduced in this study; and the hypothesis that this confusion is related to failure to click a line first or to click the correct line was corroborated.

Many students said that they valued the program because of its fast feedback, the undo feature, and the automation of box-drawing. Few referred to the justifications, however.

Steps that created sudden enlargements in the size of the proof, or introduced unknowns, bifurcations or "inscope" were often treated as indicators of error. Checking that it would be possible to prove a later line from earlier lines (by interpreting the logical connectors) was not a common strategy.

This study took place some five months after the end of the teaching, and some students were revising for the exams more than others, so it would be difficult to conclude much about how easy students found it to recall what they had learned previously. Nevertheless, when a similar proof segment was proved in separate proofs in this study, nearly all students when asked said that they had not noticed the similarity. Indeed, sometimes students even had more difficulty with the second attempt at a similar proof segment than the first.

Implication and Conjunction tended to be seen as the easiest topics; Disjunction was next; Negation and Quantifiers were held in about equal dread.

Students said that when they needed help with logic, they tended to seek help from the lecture notes, the lab assistants, fellow students, and their tutors. The ItL Jape manual or logic textbooks were not consulted.

The typical pencil-and-paper perception of a proof as a written, linear sequence of logical formulae contrasts with a possible perception under Jape that a proof is a set of simplifications of the conclusion and premises. Rules feature as technical warrants for lines when using pencil-and-paper, but they are "applied" in Jape to generate lines and justifications automatically. One effect of this difference in perception is whether students focus on complicated formulae or on missing justifications when deciding how to proceed.

5.4 Explaining student behaviour in terms of prerequisite knowledge of the rules

5.4.1 Design needs - four user groups

Four groups of users can be considered, with respect to their prior knowledge of the rules:

(1) Those who know the name of the rule they want to apply.

(2) Those who know how they want the transformed proof to look, but are less sure about the name of the rule that achieves this transformation.

(3) Those who have a partial understanding of the rules and are trying to work out how they can be used.

(4) Those who have never met the rules before.

Group 1 and Group 2 students may already have some experience of tackling paper proofs before using Jape; Group 3 and Group 4 students have not. In fact, Group 4 students are not target Jape users at all.

Group 1 students - the nominalists - know the name of the rule, but are not necessarily aware of what the effects of the rule might be. While they may sometimes be surprised by the effects of applying a rule, such surprise is not automatically to be taken as indicative of error; only if further reasonable moves are blocked would error be suspected. The strategy for choosing the rule would be the assumed culprit.

On the other hand, Group 2 students - the causationists - know what they expect to see, and would suspect error if they did not see it. The name of the rule is of secondary importance. Typically, when such students have constructed the proof on paper, they ask themselves as they write down the justifications "What is the rule that describes the step I've just carried out here?", rather than (say) writing down the justifications and trying to remember how the relevant rule works. Group 2 students therefore have a difficulty with Jape in that they are more comfortable carrying out transformations on a proof *without* being forced to use a named transformation.

Note that this classification of students into Group 1 if they know the name of the rule they want to apply, and Group 2 if they know the step they want to apply may not be applicable across all rules. For example, student might know "→E" as a named rule, but not know the name of the rule that justifies the conclusion P from the hypothesis P∧Q. Nevertheless, these user groups can serve as broad categories that might help in interpreting student behaviour.

Sita is clearly a Group 2 student. She knows what she expects to see: "Let's just draw the box - we're used to that"; and "If there is a choice between a complicated [looking] rule and an easy rule, I will go for the easy one first.". She knows that a particular step is wrong because the contradiction that would be required would look too complicated. She is also a forward reasoner (although not all Group 2 students are): "Assume P" she says on several occasions, and claims at one point that "You can assume anything you want" before realising that this cannot be the case: "Well not anything you want…". A forward reasoner often wants to "open an assumption box", without appreciating which rule that justifies that box. A Group 1 student cannot therefore be a forward reasoner. Sita expresses dislike of tautologies, which is typical of forward reasoners because there are no premises from which to reason forwards.

Sita is clearly not a Group 1 student. She tends to refer to rules by name only infrequently, and surprise at the effects of applying a rule tends for her to be indicative of error. Hence she pays close attention to the visual aspects of proofs: she immediately notices that Proof 21 is the previous conjecture "backwards", later she says "*Where* did you get that from?" (emphasis added); she asks to "see" things again; "What happened there?" she immediately enquires when the proof bifurcates (whereas other students might undo, or express annoyance or confusion); when working out how she wants to proceed she says not that she wants to think, but that she wants to *look* through the proof so far; she always notices whenever Bibreel fails to select the underscore in front of the unknown; and she is clearly intrigued as to why Jape appeared to show superfluous brackets. She points at the screen a great deal.

Many of Sita's difficulties - as with all Group 2 students - are associated with getting the interface to produce the effects she wants to see: "Which one do you click on?"; later she asks Bibreel whether he wants to highlight something else; "If we make an assumption… How… How are you supposed to…?". Moreover - again as with all Group 2 students - she has to find a way of remembering what rules correspond to the visual effects: "I just have to, basically, you know, try and remember the rules...".

In sharp contrast, Bibreel does not appear to be a Group 2 student. He rarely makes explicit reference to the visual, except insofar as subsequent actions look unpromising. When Sita uses the word "see", it is very often literal; whereas Bibreel tends to be figurative. Sita says "Shall we just leave it, or shall we just choose 'Done' and *see* if it works?" (emphasis added). Bibreel says "I've kind of seen it.", so Sita asks him, "What does it look like?".

However, it is not clear that he is a Group 1 student. Early on he has his own vague, idiosyncratic terminology for some of the rules, and makes mistakes in referring to them. There is evidence of forward reasoning - on two occasions he suggests that that he needs to make an assumption; although generally he seems to show flexibility as to rule direction.

Bibreel's language and actions - at least at the start - seem to situate him as a Group 3 student, as evidence by his attempt to second guess the intentions of the person who set the conjectures, his acceptance of the outcome of rules even when he is not sure what will follow, his refusal to see that the fact that one conjecture was the converse of the other may be of relevance, his hit-and-miss approach to ¬I, his comment "You can tell after a while when the method is going to work.", and his repeated reference to "instinct" as the best guide to action.

He has a vague understanding of some of the rules, and is clearly trying to use the feedback provided by the program in order to understand them better. In contrast to Sita, Bibreel is focussed on assessing the fruitfulness of different *actions* rather than on producing expected visual forms. He has explicit strategies for using particular rules, such as «when ∨ is at the top, you do ∨E», that he expects to improve (he points out that he doesn't know *what should be done* when ∨ is at the bottom, and that he doesn't know *what to do* in order to split up a negation), and the role of the feedback from a rule application is to help improve these strategies rather than to indicate whether the rule produced the instantaneous visual effect that was desired. It is perhaps emblematic that when stuck at one point, Bibreel says "Don't know what to *do*.", whereas Sita says "Can't *see* what's wrong with it." (emphasis added in both cases).

By the end of the session, however, Bibreel shows more confidence with the rules. He uses the rule names frequently and accurately. He, not Sita, is the one directing the activity of clicking lines and rules, and indicating value of the ellipsis. His proof strategies are more successful than at the start. He ignores all attempts by Sita to introduce arbitrary assumptions, for example, and has a robust strategy for choosing the contradiction in ¬I. He asks questions such as "Can we prove P?". His concern is always the long-term fruitfulness of a rule rather than the immediate effect: "It can't possibly want me to do *all* this.", he says, and "it's just you go from one step to another to another. And things are just making you go in that direction. You're just flowing. And you don't necessarily *know* why. There's a lot of gut feeling; there's a lot of instinct". The difficulty he reports is not - as with Sita - remembering what the rules do, but "little tricks and things. I mean, you see, for me, there was a consistent pattern where I kept falling through the trap". In other words, the feedback of the program was used to debug his proof strategies rather than simply to indicate conformity with a linear paper proof.

It appears then, that Bibreel has turned from a Group 3 student at the start into a Group 1 student by the end. Sita, however, is still struggling to produce by using the rules the visual output from forward steps.

5.4.2 How the different user groups learn from Jape

Before attempting to use this distinction (in terms of the knowledge of the rules) to explain behaviours or to predict the effects of changing the program's interface or functionality, we should first attempt to analyse the strategic-theory problem situation that each group faces.

Group 1 students

For each stage of a paper proof, a Group 1 student (i) chooses a rule to implement (using what might be called a "rule-choice strategy"), (ii) implements the rule (using a "rule-implementation strategy"); and (iii) justifies new lines (using a "justification strategy"):

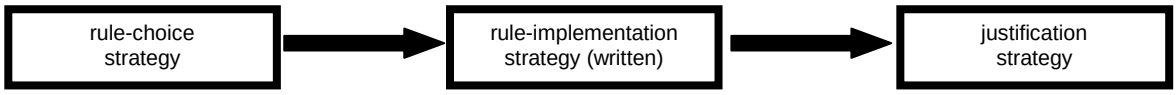


Figure 138: Paper proof – Group 1 students

How is the situation different under Jape?

Jape takes care of the justification strategy, and makes the rule-implementation strategy much easier. It also provides feedback that can be used in an assessment of whether the new proof constitutes a movement in the direction of proof completion (what might be called a "progress-assessment strategy"), so as to debug their rule-choice strategy:

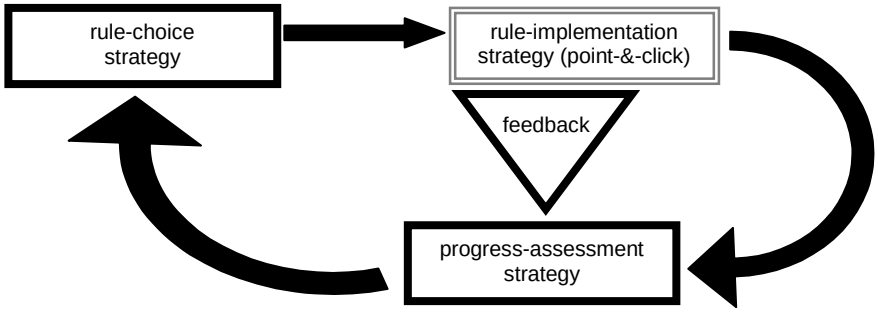


Figure 139: Jape proof – Group 1 students

In the long-run, therefore, one might conjecture that these Group 1 students are unlikely to develop new written rule-implementation or justification strategies when using Jape (and that they may even eventually forget how to do these things on paper); but that they are likely to be able to focus on developing rule-choice strategies in more difficult proofs.

However, it is very important to notice that this situation has been simplified quite a bit for the purpose of illustration.

Firstly, for example, it is likely that the rule-choice strategy consists of something like: draw up a mental shortlist of possible rules for the current proof; and then for each rule work out the effect on the proof of the rule (a "rule-effect strategy") and assess whether the new proof constitutes an improvement. In other words, the rule-choice strategy may in fact depend on a vaguer version of the written rule-implementation strategy and on some sort of progress-assessment strategy. So when using Jape, students may, after all, improve rule-implementation and progress-assessment strategies.

Secondly, if, when using Jape, one is developing a rule-effect strategy, this is also likely to assist the development of a justification strategy.

Thirdly, it should not be assumed that paper proofs offer no opportunity for feedback. Students working on paper proofs may realise, having implemented or justified a rule, that the rule-choice strategy was in error, and so may improve the rule-choice strategy. This learning mechanism depends of course on having accurate written rule-implementation and justification strategies, which Jape's mechanism does not.

Fourthly, recognising that a strategy produced the wrong result and correcting the result is not the same thing as discovering the error in the strategy and correcting the strategy.

Finally, it is an important point that the feedback provided by Jape can be used in principle to debug a variety of strategies - rule-choice, rule-effect, rule-implementation, and progress-assessment. But it may not be obvious to the novice which strategy has gone wrong. Nevertheless, the assumption for Group 1 students is that they have reasonably accurate strategies all round, but want to refine their rule-choice strategies in more difficult proofs.

Group 2 students

For each stage of a paper proof, a Group 2 student (i) chooses a *step* to implement (using a "step-choice strategy" - note that we refer here to a "step" rather than a "rule", because a Group 2 student's strategy for constructing proofs does not initially require them to know the name of the rule); (ii) implements the step (using a "step-implementation strategy"); (iii) finds the rule corresponding to the step (using a "step-name strategy"); and so (iv) justifies new lines (using a justification strategy, as for Group 1 students).

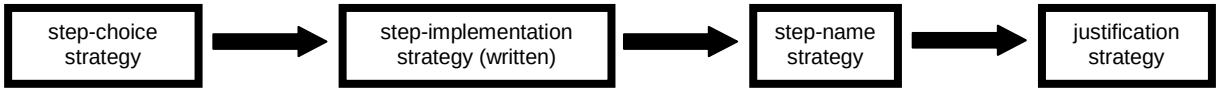


Figure 140: Paper proof – Group 2 students

What happens to this proof-construction strategy with Jape?

Jape provides feedback that can be used in a progress-assessment strategy. The program also takes care of the justification strategy, but it does not allow the user to implement a step without selecting the corresponding rule. Hence with Jape, the step-name strategy has to be used *before* using the step-implementation strategy, unlike paper. For example, a Group 2 "forward reasoner" would naturally write down an assumption before having decided what rule will allow the assumption; and so will face difficulties when Jape does not allow this.

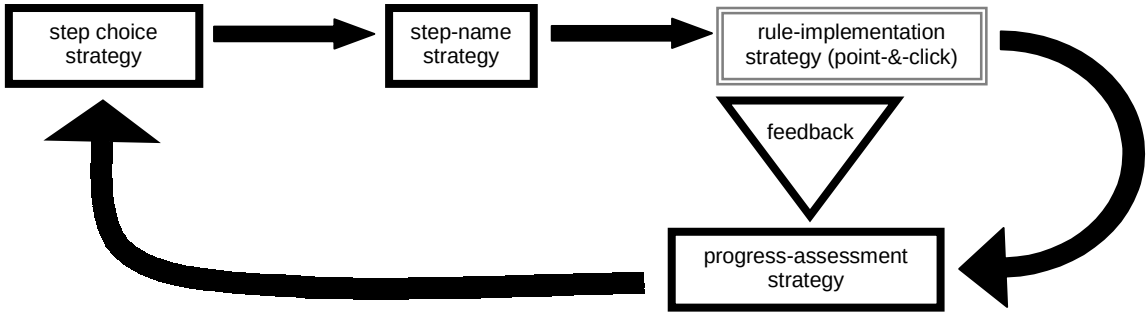


Figure 141: Jape proof – Group 2 students

Because they know at least what *step* they want to carry out, if their step-choice strategy is reasonably accurate, they can in principle compare the output of their chosen rule with what they wanted to do and so quickly debug their step-name strategy. Moreover, if both their step-name and progress-assessment strategies are reasonably accurate, they can quickly debug their step-choice and rule-implementation strategies, and so become Group 1 students (although it is not clear what role might be played by rule-effect strategies here).

However (and this is a crucial difference from Group 1 students), if *both* the step-name and step-choice strategies are rudimentary, they will be attempting to use the feedback provided by Jape to debug the two at the same time. This may be a serious hurdle to progress until their step-name strategy is sufficiently developed.

Group 3 students

For each stage of a *paper* proof, Group 3 students will make best (but idiosyncratic) use of their limited knowledge to progress. Since they receive no feedback except through comparison with lecture notes and comments from tutors, it is difficult to predict how their strategies improve, or even whether they will end up as Group 1 or Group 2 students.

However, when they start using ItL Jape, this idiosyncratic process is dramatically transformed. In order to progress, they must select a rule and use the feedback to determine if it was a good choice. Hence it is likely that *if* they are successful in learning from ItL Jape they will turn into Group 1 students rather than Group 2 students. But it difficult to see how they can successfully learn from ItL Jape without external assistance (e.g. lecture notes or tutor), because not only do they rely on the feedback to help them choose a rule to implement, but they also have poor progress-assessment strategies (except for very simple cases). Moreover, because at the start they do not have a sophisticated rule-choice strategy (perhaps just symbol-matching), they often have at least four possible rules to check (introduction and elimination rules for a premise and a conclusion) - perhaps more if there are multiple premises or the possibility of proof by contradiction.

5.4.3 Explaining episode 15 in terms of user groups

The model outlined above can go some way to provide an explanation for episode 15. Sita is not only a forward reasoner, but also has only rudimentary step-name strategies; so she struggles to produce the visual output she expects to see. Consequently, she fails to debug her step-choice strategies.

In contrast, Bibreel starts off with poor rule-choice and step-choice strategies, but is able to use the feedback from rules to assess progress. The ellipsis would appear to be important here, but crucially, the accuracy afforded by Jape allows him to focus on developing strategies for checking provability, for determining the ¬I contradiction, for backwards reasoning in the case of →I, for determining when ¬E¬I is useful, and perhaps for when attending to justifications is useful. He ends up a Group 1 student, and appears to be on his way to success.

Their logic exam was a few days after this interview, and the difference in exam scores may be instructive. Having scored slightly below average in both Logic1 and Logic2, Bibreel scored well above average in the exam. Sita,

meanwhile, scored slightly above average in Logic1 and scored very poorly in Logic2; her mark for the exam was not as poor as for Logic2, but it was still below average. She just scrapped a pass overall.

The difference in their success in learning from Jape in episode 15 may of course be more indicative of the interplay of their personalities in the interview than of a general trend that applies to all Group 2 and Group 3 students.

However, we have seen earlier that there were crucial distinctions between the two of them. Firstly, Bibreel made the decision "I'm actually going to see now using Jape rather than going about what we did [on paper]" that Sita did not - her aim was to reproduce on screen the steps she used on paper. Secondly, Bibreel was prepared to abandon forward-fixated reasoning quickly; whereas it is not even clear by the end whether Sita had abandoned it. And thirdly, Bibreel indicated that he found Jape valuable because the program is fast and accurate "And you can trial-and-error a lot."; whereas Sita appeared not to value so much the chance to experiment - The value of Jape for her in indicating straightaway that she has "done it wrong" was arguably outweighed by the struggle to produce expected output by forward steps.

In summary, Jape would appear to be more supportive of those who are willing to explore than of those who want to reproduce paper proofs.

5.4.4 Explaining inefficient mouse activity in terms of user groups

Inefficient mouse activity was repeatedly observed in the Reflection Study. Assuming that it is not mouse doodling, one interpretation is that the student are not sure which rule they want and are using the mouse pointer as a visual aid to help them mentally "cross out" potential choices. The students would therefore belong to either Groups 2 or 3. However, the activity is arguably too quick to be playing such a role. Moreover, very often the mouse shuttled back and forth between the introduction and elimination sections of the menu.

Another interpretation is that they know what rule they want, but are just having difficulty finding it. The students would therefore belong to Group 1.

A further interpretation is that the students are using "symbol-matching" as a fallback strategy for rule-choice. For example, they are looking for a symbol such as "→" or "¬" because of a hunch about the most productive formula to work on. But they are not entirely sure whether it is introduction or elimination and so prevaricate between the two (and, after all, they have got the choice wrong before, and it is easy to say one when you mean the other, and, in any case, "introduction backwards" eliminates). This behaviour situates them as Group 3 students.

Whatever the case, it would appear that the structure of the rules menu is important for supporting students' rule-choice strategies.

5.4.5 The impact of the rules menu on different user groups

If Group 2 or Group 3 students are "symbol-matching" to find rules it might be better to go with expectations and so minimise unnecessary mouse pointer movement by classifying the rules by symbol. That is:

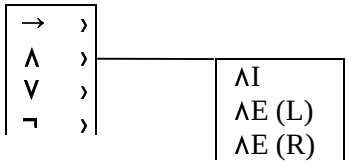


Figure 142: Rules menu classified by symbol

(→ would have the sub-menu "→I →E", ∨ would have the sub-menu "∨I (L) ∨I (R) ∨E", ¬ would have the submenu "¬I ¬E")

rather than:

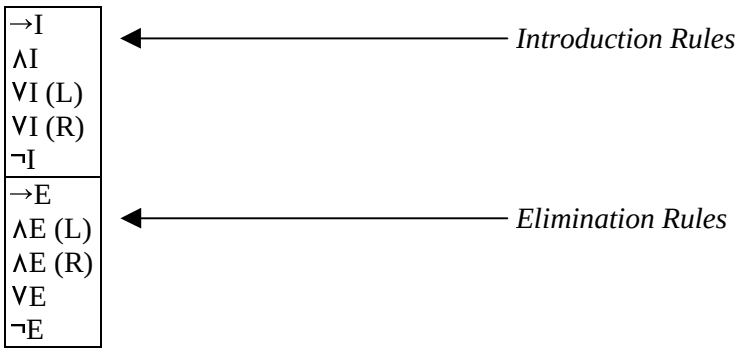


Figure 143: Current rule menu (quantifiers omitted for clarity)

However, one student objection to this might be along the lines of "Jape knows what principal operator I've selected, so why do I have to select the symbol at all?", although there is ambiguity with respect to negation.

Group 1 students - the nominalists - want the most efficient interface for finding and applying the rule they have chosen, so they might want the rule menu to be classified by introduction/elimination, as now, because it is a simple binary choice that divides the search space of rules into two similarly-sized parts while making reference to the rule names.

Group 2 students - the causationists - might prefer the rule menu to be classified by forwards/backwards rather than introduction/elimination, because although again it is a simple binary choice that divides the search space of rules into two similarly-sized parts, it makes references to the *effect* of a step rather than to the name of the rule. In Thomas Green's scheme of cognitive dimensions, the current requirement to make the introduction/elimination choice *before* choosing the direction might be seen by such students as "premature commitment". This might explain the confusion reported by some students about the program applying a rule in the direction opposite to the one they expected.

Figure 144: Rules menu organised by forward/backward steps

The introduction/elimination rules menu shows the five introduction rules versus the five elimination rules, any of which might, as far as the novice student would be concerned, be applied forwards or backwards, making a total of 20 possible steps. The forward/backward rules menu shows five "forward steps" versus six "backward steps", making a total of 11 possible steps. This classification - which almost halves the search space for most situations - relies on the fact that students nearly never require Jape to apply introduction rules forwards and elimination rules backwards. The exceptions are ∧I-forwards (which Jape doesn't do, in any case); and ¬E-backwards, which perhaps isn't required if the ¬I rule is re-jigged slightly. (Recall that ItL Jape has been designed to mirror the rules used in the logic course, and these are open to question. For example, the requirement to prove _B∧¬_B, rather than just a contradiction, can cause difficulties.).

There are number of difficult questions that have to be raised about this sort of reclassification. Firstly, of course, what happens when the student wants to carry out a step not featured here, such as an introduction rule forwards or an elimination rule backwards? Secondly, it is far from obvious whether students in Groups 1 and 3 will be clearer about the *direction* of a rule than the *change that the principle connective* undergoes.

Group 3 students - the experimentalists - might be best served by some sort of diagrammatic icons, mnemonics or animations that indicate the structure or actions of the different rules.

Is it possible to indicate the rules within the program using schematic representations in addition to the names? The difficulty is then one of capturing the sense of the rule without cluttering up the menu, which might make it harder

rather than easier to select a rule at speed. There is also the problem that if the name of the rule is bypassed completely, writing down the justifications on paper will be rather difficult.

5.4.6 The impact of increasing automation on different user groups

Jape can offer to the educator a double-clicking mechanism that enables students to simplify formulae without having to select a particular rule from the menu. A comparison of this mechanism with the approach that ItL Jape uses would be interesting, but unfortunately there is no time available in this current research for such a comparison. However, it is possible to conjecture the effect of introducing such automation on the different user groups. It would be possible to automate in more ways than just through instant application of obvious rules by double-clicking. In particular there would be different ways to handle the ambiguities of ∧E, ∨I and ¬E. Nevertheless, this particular mechanism offers scope for prediction about learning outcomes.

Group 1 students

One advantage of the double-clicking feature for Group 1 students is that rule applications that they know they want to carry out can be done quickly, leaving the students free to focus on how to construct hard proofs rather than how to apply particular rules. However, this requires them to know which is the appropriate expression to double-click, which is one step removed from their forte (knowing the names of the rules). Double-clicking would therefore, paradoxically, initially slow them down; and then (when they become familiar with the feature) they may become disconnected from their paper-based experiences of proof construction. This may in itself be no bad thing if they are sufficiently experienced to be able to remember or reconstruct their knowledge of the rule names.

Group 2 students

Automation is potentially disastrous for Group 2 students. On the plus side, forward reasoners would soon discover that the only way to "assume something" was by double-clicking on implications, thus turning themselves (at least for →I and ¬I) into flexible reasoners. However, automation would tend to militate against students reflecting on *which* rule has been applied. They would be able to construct complicated proofs, but the strategies they develop would not, in the main, transfer to paper, because step-name strategies would remain undeveloped. This hypothesis would be important to test in future research.

On the other hand, if the production of paper proofs by students is *not* a desired outcome, then automation might be appropriate for Group 2 students. The difficult question then is whether what it is that is being learned is sufficient for an introduction to logic? If paper proof skills *are* desired, then it would seem that a different feature would be needed in Jape - students would have to provide justifications for lines in response to steps chosen by the computer.

Group 3 students

Double-clicking automation provides a chance for Group 3 students to learn what sort of things can be done to a proof (and thereby begin to appreciate something of the structure of rules) and to gain experience of constructing proofs. Moreover, provided rules are not required at all, students may develop strategies for step-choice, because rules do not have to be chosen, because steps do not have to be implemented, and because progress-assessment is made easier by the reduced number of proofs that have to be evaluated. In other words, Group 3 students may turn into Group 2 students if there is automation. However, strategies for implementing steps on paper and for finding the rules corresponding to steps are likely to remain rudimentary.

5.4.7 Increasing the freedom to err

One way of making the Jape experience more productive for Group 2 users - in particular forward reasoners and those who are set on reproducing paper proofs - might be to add an option to turn off at least some of the ItL Jape restrictions that inhibit poor proof practice. For example, students could be allowed to "Make an assumption" (rather than having to initiate assumption boxes using →I, ∨E or ∃E). There would consequently also need to be an instruction to "Finish this assumption". In addition, under this option, the method of creating a formula such as P∧Q by clicking the premises P and Q and applying ∧I forwards could be allowed; this method is again "natural" for some students.

Under this option, the clarity of the important principle that arbitrary assumptions are never required would be decreased, and the interface would be more clumsy. However, it would firstly allow students to use Jape to check their paper proofs with ease and so feel more comfortable with Jape as a logic calculator that is consonant with the way they work on paper, rather than seeing it as an idiosyncratic tool that forces them to work differently (and therefore, for them, perhaps less efficiently).

There may, however, be a way of increasing both the confidence that Group 2 students have in Jape's value and their likelihood of shaking off any forward-fixated reasoning tendencies. An advice panel opposite the proof could display hints at appropriate moments in a proof. For example, should "Make an assumption" be selected, a hint could be displayed that this is not the optimal course of action. By letting the student follow the sub-optimal path the program thus allows students to attempt the conjecture again with the aim of improving their proof steps. If the hint is not taken at first, future hints could be more explicit: "That step is unnecessary - you could apply →I to line 5", for example.

This may be a more powerful way of encouraging students to appreciate the value of backwards reasoning than preventing them from poor practice completely. The advice panel could also show error messages, thus reducing the need for attention-diverting clicks to dismiss a dialogue box.

5.5 Explaining student behaviour in terms of proof strategies

5.5.1 The existence of proof strategies

The distinction between rule-specific and global strategies has already been described. Rule-specific strategies tend to help students choose which rule to apply next, either by specifying a situation and the rule or rules that might be necessary or useful in that situation, or by providing a way of deciding between a forwards rule and a backwards rule. Global strategies help students plan and debug an approach to the proof.

One strategy that Bibreel seemed to be refining in episode 15 appeared to something akin to « If no forward steps are possible, consider →I backwards. If that's not possible, consider ¬I backwards. If the conclusion is not a negation then use ¬E backwards before ¬I». Another is «When deciding on a contradiction to use for ¬I, consider first any hypotheses that have not already featured in a step». These examples illustrate that rules are sometimes involved in several strategies, that strategies can involve several rules at once, and that the combination of simple strategies into more complex strategies is a matter of personal judgement.

Interviews with students in the Reflection Study produced linguistic formulations that elaborated on similar ones noted in the Observational Study. However, these formulations are clearly ad hoc, in that success with classes of situations that was explicable in terms of strategy did not automatically mean that the student could describe with ease how these situations were being tackled.

These points about the variation in linguistic and combinatorial formulation mean that an attempt to catalogue hypothesised proof strategies would perhaps be of little use. Moreover, there is a danger that on the one hand these strategies may be misinterpreted by logicians as being unclear, misleading or false statements about the logic; and on the other hand may be misinterpreted by educators as being misguided instructional devices. Of course they are neither; they are models that attempt to explain students' behaviour

Nevertheless, with these caveats in mind, some strategies stood out in the Reflection Study as being fairly common:

Rule-specific strategies for individual rules

→I backwards	«If the principal operator in the conclusion is →, break up that conclusion by using →I backwards - it creates an assumption box that often allows forward reasoning» *Jape needs you to click the conclusion before applying the rule*
→E forwards	«If the principal operator in a hypothesis is →, look to see if the term on the left of the → is a hypothesis or provable - you can then substitute it into the hypothesis by using →E forwards, to get the term on the right of the →» *Jape needs you to click the hypothesis containing the → before applying the rule*
∧I backwards	«If the principal operator in the conclusion is ∧, break up that conclusion by using ∧I backwards - it turns the two terms in the conclusion into lines to be proven» *Jape needs you to click the conclusion before applying the rule*
∧I forwards	«Two lines can be combined into a single line by using ∧I forwards» *On paper only - not Jape*
∧E forwards	«If the principal operator in a hypothesis is ∧, you can break down the hypothesis into separate lines by using ∧E forwards» *Jape needs you to click the hypothesis before applying the rule, and needs to be told whether you want the left-hand side of the conjunction or the right.*
∨I backwards	«If the principal operator in the conclusion is ∨, and one of the two terms is a hypothesis or provable, then turn that term into a line by using ∨I backwards» *Jape needs you to click the conclusion before applying the rule, and needs to be told whether you want the left-hand side of the disjunction or the right.*
∨E forwards	«If the principal operator in a hypothesis is ∨, you can break it into two separate cases by using ∨E forwards - if you can prove the same conclusion from each case then you can write that conclusion as a line» *Jape needs you to click the hypothesis before applying the rule.*
¬I backwards	«If the principal operator in the conclusion is ¬, attempt to prove it by contradiction by using ¬I backwards. When deciding on a contradiction to use, consider first any hypotheses that have not already featured in a step.» *Jape needs you to click the conclusion before applying the rule, and will need to be told later what the unknown is to represent*
¬E forwards	«If a hypothesis has ¬¬ at the start, they can be removed by using ¬E forwards» *Jape needs you to click the hypothesis before applying the rule.*
¬E backwards	«If no useful forward steps are possible, attempt to prove it by contradiction by using ¬E backwards, followed by ¬I backwards» *Jape needs you to click the conclusion before applying the rule.*
∀I backwards	«If the principal operator in the conclusion is ∀, attempt to prove it by using ∀I backwards - this rule introduces a fresh variable, in a scope box, and if you can prove something for this fresh variable then it is proved in general.» *Jape needs you to click the conclusion before applying the rule.*
∀E forwards	«If the principal operator in the hypothesis is ∀, it can be applied to an existing variable by using ∀E forwards.» *Jape needs you to click the hypothesis before applying the rule, and it is also a good idea to pass the variable as a parameter to the rule by using text-selection.*
∃I backwards	«If the principal operator in the conclusion is ∃, prove it from a specific case by using ∃I backwards. » *Jape needs you to click the conclusion before applying the rule, and it is also a good idea to pass the variable as a parameter to the rule by using text-selection.*
∃E forwards	«If the principal operator in the hypothesis is ∃, conclusions can be drawn from it by using ∃E forwards - this rule introduces a fresh variable, in a scope box, and if you can prove something for this fresh variable then it is proved outside the scope box. » *Jape needs you to click the hypothesis before applying the rule.*

Rule-specific strategies that choose between rules

«Try →I backwards before →E forwards»
«Try ∨E forwards before ∧I backwards, ∨I backwards and ¬I backwards»
«Use ∀I backwards before ∀E forwards»
«Use ∃E forwards before ∃I backwards»
«Use ∃E forwards before ∀E forwards»
«Try →I backwards before ∀E forwards»

Rule combination strategies

¬E¬I∧I	«In the case ¬A⊢B, convert the proof into ¬B⊢A if A is easier to break up than B, by using ¬E¬I∧I - use A∧¬A as the contradiction. Similarly, you can convert A⊢¬B into B⊢¬A, ¬A⊢¬B into B⊢A, and A⊢B into ¬B⊢¬A »

Global proof strategies

	«Look for the most complex line in order to decide on a focus for attention»
	«Look at the gaps in the justifications to decide on a focus for attention»
	«Look at the lines either side of the ellipsis to decide on a focus for attention»
Symbol matching	«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»
Break down & build up	«Break down complex premises into components using elimination rules, and then build up the components into conclusions using introduction rules»
S1 (from lecture notes)	«Begin by asserting the premises»
S2 (from lecture notes)	«Reason forwards from the premises»
S3 (from lecture notes)	«Reason backwards from the conclusion»
S4 (from lecture notes)	«When all else fails, assume the complement of the sentence you are trying to prove and aim to derive a contradiction»
Check back	«When reasoning backwards, check if the lines produced are provable from the premises»
Check forward	«When reasoning forwards, check if the lines produced are useful in obtaining the conclusion»
	«If stuck on a conjecture with no premises, try to think of a previously-proved theorem that could be applied, so as to allow forward reasoning»

No evidence was found of strategies concerned with efficiency of proofs, as distinct from strategies concerned with obtaining a proof at all.

5.5.2 The development of strategies

A distinctive feature of Jape users was that a strategy akin to «Break up implications in the conclusion» typically became almost automatic very quickly; whereas when using pencil-and-paper prior to Jape, the less reliable strategy «Make an assumption.» tended to predominate.

What aspects of ItL Jape's interface encourage the development of robust strategies? It must clearly help when starting to use the program that it adopts familiar gestures, commands and visual cues from commonplace graphical operating systems, and that its proof display is just like that in the course. In addition, many students praised the opportunity for experimentation provided by the guarantee of accuracy in applying chosen rules and by the "undo" facility for when poor choices became apparent.

It is not clear yet why some students seemed more prone than others to fruitless pattern-matching trial-and-error. However, strategy-development seemed most successful when the visible effects of an action were not only sufficient to allow students to make an informed decision about the utility of the action, but were also subtle enough to place the onus of strategy-development on the student. One crucial factor would also appear to be students' *initial expectations* of the effects of a rule.

5.6 What students need to know in order to use Jape

Students need to have two sorts of prerequisite knowledge - about the interface and about the rules.

Prerequisite knowledge about the interface

A one-page "quick help" document has been prepared to give all the information about the interface that is considered necessary for students who are already reasonably familiar with the Mac or Windows operating systems, and with the local network facilities.

The document explains how to specify which conjecture they want to prove, how to apply a rule, how to undo a rule, how to indicate that a proof is complete, how to make an assumption ("You don't. You find a rule that makes it for you, such as →I backwards, ∨E forwards, ¬I backwards, or ∃E forwards"), how to indicate what "_B" should represent using text-selection, and how to pass a parameter to ∀E and ∃I using text-selection.

These instructions are a way to bypass some of the main interface difficulties experienced by students. The document does not explain how to apply theorems, or how to use the hyp rule; and does not explain the role of arguments, unknowns, variables, the ellipsis and scope boxes in the context of each rule. Users are referred to the manual for this. Most novice users can either bypass such aspects using the instructions given or else discover the aspects for themselves.

Prerequisite knowledge about the rules

Determining the logic knowledge that is required to use the program is clearly a rather difficult task. Although it seems clear that Group 3 students have less prior logic knowledge than the Group 1 or Group 2 students, it is not certain - despite Bibreel's success - that the knowledge of Group 3 students is a sufficient guarantee of productive use of Jape.

Given the success of the students who used Jape for more than a couple of hours, something rather less than a 12 week logic course is required; but given the difficulties of students such as Lewis, something rather more than just sitting through a few lectures and completing a few exercises might be needed. The widely held view of the students who experienced difficulties with disjunction and quantifiers that they needed to "learn the rules" must be taken seriously.

It can be conjectured, based on the earlier theoretical analysis of Group 3 students' learning, that this knowledge to which they refer relates to the ability to choose an applicable rule for at least simple situations (the assumption being that the ability to tackle more difficult situations would be learned through Jape) and to the ability to use the feedback provided by ItL Jape to assess whether the rule application resulted in progress towards completion of the proof; in short - *elemental rule choice strategies* and *progress-assessment strategies*.

6. 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.

7. References

Aczel, J. C. (1998) “Learning Equations using a Computerised Balance Model: A Popperian Approach to Learning Symbolic Algebra”, unpublished DPhil thesis, University of Oxford

Aczel, J. C. & Fung, P. (1999) “Evaluation of a software tool for supporting formal reasoning”, *Proceedings of the day conference of the Student Research Centre*, Institute of Educational Technology, The Open University

Aczel, J. C., Fung, P., Bornat, R., Oliver, M., O’Shea, T., & Sufrin, B. (1999) “Using computers to learn logic: undergraduates’ experiences”, *Proceedings of the 7th International Conference on Computers in Education*, Amsterdam, IOS Press

Aczel, J. C., Fung, P., Bornat, R., Oliver, M., O’Shea, T., & Sufrin, B. (1999) “Influences of Software Design on Formal Reasoning”, in Brewster, S., Cawsey, A. & Cockton, G. (Eds.) *Proceedings of the Seventh IFIP TC.13 International Conference on Human-Computer Interaction (INTERACT ‘99)*, Vol. 2, pp. 3-4, Swindon, UK, British Computer Society, ISBN 1-902505-19-0

Aczel, J. C., Fung, P., Bornat, R., Oliver, M., O’Shea, T., & Sufrin, B. (1999) “Undergraduates’ Computer-Assisted Formal Reasoning”, Paper presented at the day conference of the British Society for Research into Learning Mathematics, Feb 1999

Aczel, J. C., Fung, P., Bornat, R., Oliver, M., O’Shea, T., & Sufrin, B. (1999) “Computer Science Undergraduates Learning Logic Using a Proof Editor: Work in Progress”, *Proceedings of the Psychology of Programming Interest Group*, University of Leeds

Aczel, J. C., Fung, P., Bornat, R., Oliver, M., O’Shea, T., & Sufrin, B. (1998) “The Effectiveness of Visualisation Tools in Supporting the Learning of Formal Reasoning for Software Development”, in Oliver, M. (Ed.) *ELT98: Innovation in the Evaluation of Learning Technology: Conference Proceedings*, pp. 31-34, University of North London

Aczel, J. C., Oliver, M. & Fung, P. (1999) “Progressive Refinement of Formal Reasoning Strategies using Software for Learning Logic”, Proceedings *of the 20th anniversary conference of the Computers and Learning Research Group*, The Open University, June 1999

Barwise, J. & Etchemendy, J. (1992) *The Language of First-Order Logic*, Third Edition, Cambridge University Press

Bornat, R. & Sufrin, B. (1992-9) “Jape” (software)
Unix/Linux version available from http://www.comlab.ox.ac.uk/oucl/users/bernard.sufrin/jape.shtml
Apple Macintosh version available from http://www.dcs.qmw.ac.uk/~richard/jape

Bornat, R. & Sufrin, B. (1996) “Animating the formal proof at the surface: the Jape proof calculator”, Technical Report, Department of Computer Science, Queen Mary & Westfield College, London, ftp://ftp.dcs.qmw.ac.uk/jape/papers/

Bornat, R. & Sufrin, B. A. (1994) “The Gist of Jape”, PRG Research Monograph

Dyckhoff, R. (1987) “Implementing a simple proof assistant”, Workshop on Programming for Logic Teaching, Centre for Theoretical Computer Science, University of Leeds

Fitch, F. B. (1952) “Symbolic Logic”, Ronald Press, New York

Fung, P. & O’Shea, T. (1992) “Using Software Tools to Learn Formal Reasoning: a first assessment”, CITE Report No. 168, The Open University, UK

Fung, P., O’Shea, T., Goldson, D., Reeves, S. & Bornat, R. (1993) “Computer Science Students’ Perceptions of Learning Formal Reasoning Methods”, *International Journal of Mathematical Education in Science and Technology*, 24 (5), 749-760

Fung, P., O’Shea, T., Goldson, D., Reeves, S. & Bornat, R. (1994) “Why computer science students find formal reasoning frightening”, *Journal of Computer Assisted Learning*, 10, 240-250

Fung, P., O’Shea, T., Goldson, D., Reeves, S. & Bornat, R. (1996) “Computer tools to teach formal reasoning”, *Computers in Education*, 27 (1), 59-69

Green, T. R. G. (1989) “Cognitive Dimensions of Notations” in Winder, R. & Sutcliffe, A. (Eds.) *People and Computers V*, p. 443-460, Cambridge University Press

Guba, E. & Lincoln, Y. (1981) “Effective evaluation: improving the usefulness of evaluation results through responsive and naturalistic approaches”, Jossey-Bass, London

Hodges, W. (1977) *Logic*, Penguin

Kadoda, G. (1997) “Cognitive Dimensions Analysis of Theorem Provers”, *Proceedings of the 3rd International Workshop on User Interface Design for Theorem Proving Systems*, France

Reeves, S. & Clarke, M. (1990) *Logic for Computer Science*, Addison-Wesley

Scheines, R. & Seig, W. (1993) "The Carnegie Mellon Proof Tutor" in Boettcher, J. V. (Ed.) *101 Success Stories of Information Technology in Higher Education: The Joe Wyatt Challenge*, McGraw-Hill

Sufrin, B. A. & Bornat, R. (1995) "Designing structural induction rules for Jape". PRG Research Monograph

Sufrin, B. A. & Bornat, R. (1996) "Jape's quiet interface", Proceedings of UITP-96

Sufrin, B. A. & Bornat, R. (1996) "User interfaces for generic proof assistants part 1", Proceedings of UITP-96

Sufrin, B. A. & Bornat, R. (1997) "Displaying sequent-calculus proofs in natural deduction style", Proceedings of PTP-97

Sufrin, B. A. & Bornat, R. (1998) "User interfaces for generic proof assistants part 2", Proceedings of UITP-98

Sufrin, B. A. & Bornat, R. (1998) "Using gestures to disambiguate unification", proceedings of UITP-98

van Ditmarsch, H. (1998) "User interfaces in natural deduction programs", in Backhouse, R. C. (Ed.) *Proceedings of the 4th International Workshop on User Interface Design for Theorem Proving Systems*