Interactive disproof

Richard Bornat

School of Computing Science, Middlesex University. R.Bornat@mdx.ac.uk

Abstract. The proof calculator Jape has been extended to allow disproof, using Kripke forcing semantics. Users draw graphs to depict a situation, and Jape uses subformula colouring to signal the status of the situation and its components. The mechanism hasn't been scientifically evaluated, but it has been heavily tested in use. Some deficiencies are noted.

1 Introduction

Jape[3, 5, 6, 1] is a proof calculator. It was devised for use in education, specifically for early education in formal proof, and for running lightweight experiments with novel logics in formal methods research. It has a claim to be novel in that the calculator is designed to support the user interface rather than the other way around.

Recently I had experience of using Jape as the foundation of a first course in logic (until then my experience had been limited to advising others and observing[2] the results). Jape's formal proof mechanisms had been overhauled, as reported elsewhere[4], to increase its usability. In the first year of use, those improvements certainly seemed to be helpful. Stung by the criticisms of colleagues, however, I decided also to try to teach some model theory. Since constructive proof is so much more straightforward than classical proof (it really is!), my course focussed on constructive proof. It seemed only right that I should tackle constructive model theory.

The most fruitful way, I imagine, of incorporating model theory into a course on proof is to treat it as a means of disproof. If you can show a counter-example in the model, and if the logic is sound (ours is, but proof of soundness is not a part of the course) then a formal proof is impossible. You can even use a failed search for a constructive proof to guide the search for a constructive counter-example. You can have fun, in those cases where a classical proof and a constructive counter-example exist, comparing proof and counter-example of contentious examples.

All of this was done, for one year, on blackboard and on paper. The contrast between the ease with which most students learnt from Jape-supported proof search, where the machine was accepted fairly readily as an accurate and impartial arbiter in difficult cases, and the pain they experienced with paper-and-pencil counter-examples, where the only arbiters were fallible teachers, was stark. Constructive counter-models are graphical constructs with very simple rules: it was plain that machine support might be provided, and so I attempted it.

2 A scant introduction to Kripke semantics

Classical logic is based on the notion of truth. A properly formed assertion is either true or false, independently of our understanding of it. The corresponding 'method of truth tables', enumerating all possible states of the support for an assertion, is familiar to most computer scientists, at least for the propositional case and for the purposes of designing hardware. Kripke's constructive model using possible worlds[8, 7], though graphical and very accessible, is less well known. I therefore give a very brief introduction, at the level of engineering mathematics rather than foundation.

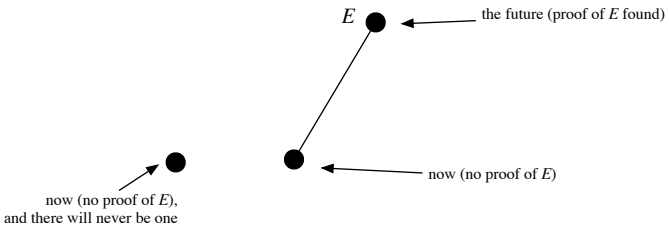

Fig. 1. Two alternative views of the present

Constructive logic is based on the notion of proof. A properly formed assertion can be proved or disproved or there can be no evidence either way. (It's not a three-valued classical logic – it's odder than that[7].) In contrast with classical logic, then, it's reasonable to consider a distinction between 'now' and 'the future': we might have no proof of assertion E now, but one might be found tomorrow, as recently was the case for Fermat's Last Theorem, and as we might hope for Goldbach's Conjecture. In the semantics the two possible situations can be shown by alternative diagrams, as in figure 1. The first (left-hand) diagram describes a situation in which there is no proof of E, and none will ever be found. The second (right-hand) world describes a situation in which there is no proof of E now, but one will be found. We don't know, when we don't have a proof of E, which of these worlds we are living in, so it might well be the second. Since constructivists interpret $\neg E$ as 'there will never be a proof of E', the second world denies both E and $\neg E$, and thus the classical law of excluded middle. On the other hand, a diagram with just a single world and no future development, like the left-hand one above, corresponds to the classical model.

Jape supports the drawing of these possible-world diagrams. The natural deduction encoding described here uses the definitions of figure 2 where $w \geq w'$ means you can travel from w to w' by following upward lines in a diagram. The glory of this collection is the $\rightarrow$, which captures beautifully the notion that I am forced to accept $A \rightarrow B$ if, however things might develop, whenever I'm forced to accept A then I'm also forced to accept B.

$w \vDash A \wedge B$ iff $w \vDash A$ and $w \vDash B$
$w \vDash A \vee B$ iff $w \vDash A$ or $w \vDash B$
$w \vDash A \rightarrow B$ iff for every $w' \geq w$, if $w' \vDash A$ then $w' \vDash B$
$w \vDash \neg A$ iff for no $w' \geq w$, $w' \vDash A$
$w \vDash \forall x.P(x)$ iff for every $w' \geq w$ and every i, if $w' \vDash$ actual i then $w' \vDash P(i)$
$w \vDash \exists x.P(x)$ iff for some i, $w \vDash$ actual i and $w \vDash P(i)$

Fig. 2. Constructive forcing semantics of natural deduction

3 Drawing counter-examples

The assertion $(E \rightarrow F) \vee (F \rightarrow E)$ has no constructive proof. An attempt to prove it using constructive rules in Jape gets stuck quite quickly, as illustrated in figure 3. There is still a constructive rule which is applicable – from contradiction conclude F – but it's pointless to try it because you can't generate a contradiction from E. The experienced counter-example generator recognises from the structure of the stuck proof that the counter-example will probably involve a world at which there is a proof of E but no proof of F.

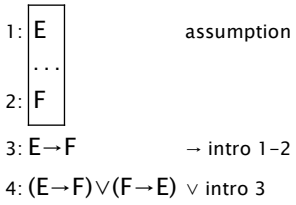

Fig. 3. A stuck constructive proof

Choosing Disprove in the menus splits the proof window into two panes and shows a minimal semantica situation, as illustrated in figure 4. The red-ringed blob in the upper pane is the currently-selected world; the assertion below it is coloured to show the status of its components in that world. Atomic assertions E and F are black, to show that they are not forced. The arrow in $E \rightarrow F$, like the brackets surrounding that subformula, is coloured violet to show that the subformula is forced – trivially, because E appears nowhere in the diagram. The arrow and brackets of $F \rightarrow E$ are coloured similarly, for similar reasons. The disjunction connective, between the brackets, is violet because $F \rightarrow E$ is forced, and the whole assertion is underlined in violet to make clear that it is forced. A counter-example demonstrates that an assertion does not always hold by showing a situation in which all the premises are forced but the conclusion is not. This is not yet a counter-example.
A new world can be created by option-dragging the base world, producing figure 5. Note that the new world makes no change in the colouring,

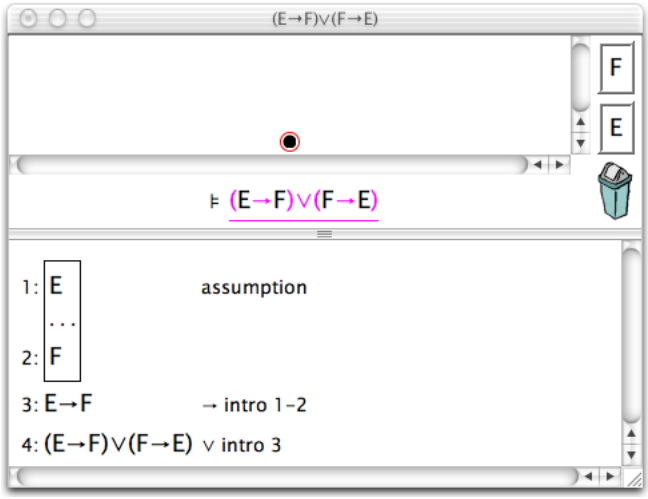

Fig. 4. The beginning of a disproof attempt

because it introduces no future developments. Then E can be forced at the new world by dragging the E tile from the right-hand side of the window and dropping it on the new world, producing figure 6. Now the colouring changes: $E \rightarrow F$ is no longer forced, because there is a reachable world at which E is forced and F is not. But $F \rightarrow E$, and therefore the whole assertion, is still forced for the same reason as before.

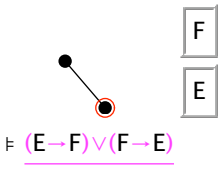

Fig. 5. An extra world

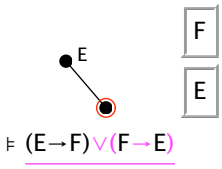

Fig. 6. A labelled extra world

At this point the experienced counter-example generator knows just what to do to complete the disproof: do for F what was just done for E. That's justified because the first proof step picked out the left-hand half of the disjunction, and we might as easily have picked the right, giving figure 7. Previously, to make a new world I dragged a new one out of the base world. That is the right thing to do again, but it is no longer the only thing to do, so I illustrate the effect of alternative choices. First, the easy way and a correct move: dragging a new world out of the base and dropping F onto it completes a disproof, signalled in figure 8 by the fact that the assertion is now *not* underlined.

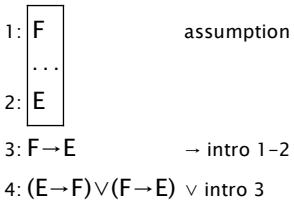

Fig. 7. Stuck again

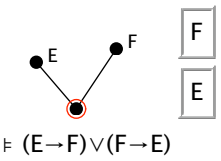

Fig. 8. The completed disproof

The other, more laboured, route to a counter-example starts by extending the E world as in figure 9. Jape enforces monotonicity when the new world is created by including E in the new world. Note that the colouring and underlining of the assertion is unchanged from the single-E situation. Then when F is dropped on the new world to give figure 10 we find that $F \rightarrow E$ is still forced, because in the top world, the only world in which F is forced, so is E. This situation isn't a counter-example.

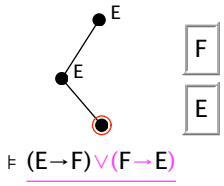

Fig. 9. Extending the upper world

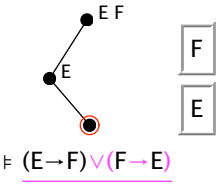

Fig. 10. Not a counter-example

It's possible to go forward in at least two ways. One way is to undo back to the two-world situation and do the right thing from there (Jape applies undo and redo to the last pane – proof or disproof – that you clicked in, rather than using an overall interaction history). Alternatively, you can modify the current situation by deleting parts of it. Monotonicity stopped us making a world with just F in it, so first I decide to get rid of the link I just made (but not the world): press on the link from the E world to the E, F world and drag it to the swing-bin in the bottom right of the disproof pane (the line stays attached to its endpoints and drags rubber-band fashion till it is dropped in the bin, which lights up, as it should, to receive the undesired object – but it's hard to capture that moment!), giving figure 11. Option-dragging the base world onto the isolated E, F world makes a new link, giving figure 12. Finally, throwing the undesired E in the bin takes us back to the counter-example of figure 8.

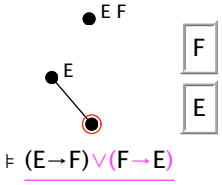

Fig. 11. An isolated world

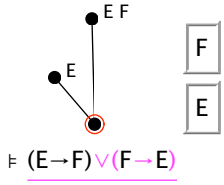

Fig. 12. A different connection

4 Exploring diagrams

The definition of $A \rightarrow B$ – where A is forced, B must be forced as well – is the glory of Kripke's model. It's also quite hard to grasp at first. I give some assistance to novices by allowing them to experiment with different situations, and more by allowing them to explore a situation – counter-example or not – to see where particular subformulae are forced. Figure 13, for example, is the result of clicking on one of the worlds of figure 10. The colouring of the assertion is now not as it was when the base world was selected. Occurrences of E are now violet, but F is still black (atomic assertions are forced if they are present, not forced if absent). The black arrow shows that $E \rightarrow F$ is not forced, and the colouring of the basic assertions shows that that's because E is forced and F isn't; similarly, it shows why $F \rightarrow E$ is forced. The disjunction, which is the overall assertion, is still forced because one of its components is forced.

That situation shows the novice why $E \rightarrow F$ isn't forced in the base world – it must hold everywhere above the base world, not just in one place, and the intermediate world contradicts it. Clicking on the top world makes the same point more strongly, producing the classical single-world evaluation of figure 14. Now everything is forced: both the atomic assertions, both the implications and the disjunction. Implication, in this semantics, is a promise rather than a static property, and a promise broken in one world is broken in all the worlds below it. Although $E \rightarrow F$ holds in the top world, and it would hold in the root world if it didn't have descendants, it doesn't hold in the intermediate world and that's enough to say that it doesn't hold in any world which has that one as a descendant.

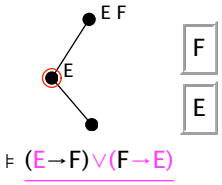

Fig. 13. Another evaluation

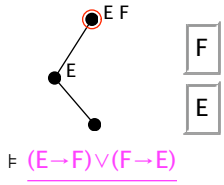

Fig. 14. A classical evaluation

Dragging and dropping is extensively supported in Jape's treatment of constructive disproof. You can drag worlds onto worlds or lines, and lines onto worlds. You can drag anything – worlds, labels, lines – into the swing bin (it's the most straightforward mechanism for deletion, vastly easier to use than a "select; delete" technique). Using drag and drop, plus an occasional focus-click, for every action in the proof pane is a worthwhile simplification, a reasonably quiet interface.

5 Working with quantifications

Forcing semantics of quantification relies on named individuals inhabiting a universe. Adding individuals to the logic, via the pseudo-assertion actual <*name*>, makes it possible to give a much more coherent account of Jape's proof capabilities. Using individuals in disproof requires a little care, and Jape is not yet completely developed in this direction.
Consider, for example, the assertion

$$\text{actual}\, i,\ \forall x.\,(R(x) \vee \neg R(x))\,,\ \neg\forall y.\neg R(y)\ \vDash\ \exists z.R(z)$$

It looks unexceptionable: in a non-empty universe (there is at least the individual i), where R is a decidable property, not false everywhere, there must be some individual with property R. But it's one of the entries in Jape's "Classical conjectures" panel, so even the novice can guess that it must have a constructive disproof as well as a classical proof.
The proof, shown in figure 15, is easy once you realise that the decidability hypothesis was inserted as a sop to the constructivists, and even the non-emptiness of the universe is nothing to do with anything. But the constructive attempt, though it uses all the premises and explores more avenues, gets stuck as in figure 16 because it can't make the same first move.
Never mind! The stuck proof tells us what to do: make a world with an individual i (line 1), another with $i1$ (line 7), don't have $R(i)$ (line 6), but do have $R(i1)$ (line 8). Jape's starting point for a disproof, shown in figure 17, colours actual i black, because it doesn't occur at the root world, underlines the second premise, which is forced trivially because there are no individuals in the diagram, and doesn't underline the third for similar reasons. The conclusion then isn't forced because there is no individual to stand witness. The tiles on the right allow us to make worlds involving the only individual (i) and the only predicate (R) in the assertion. They won't be enough, but we can make a start.
Recall that to produce a counter-example we must make a situation in which all the premises are forced and the conclusion is not. So the first step must be to force actual i in the root world (figure 18). Observe that actual i is now forced, and both $\forall$s are too, because $\neg R(i)$ is now forced. The $\exists$ is still black. And notice that the formulae inside the quantifications are grey: a new kind of colouring.[1] Greyness signifies that a formula isn't strictly relevant to the status of the assertion, and that's

[1] For those of you reading in black and white, as they almost used to say on UK television, the grey bits look just like the violet bits. Sorry!

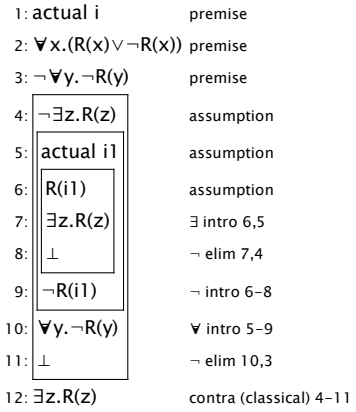

Fig. 15. A classical success

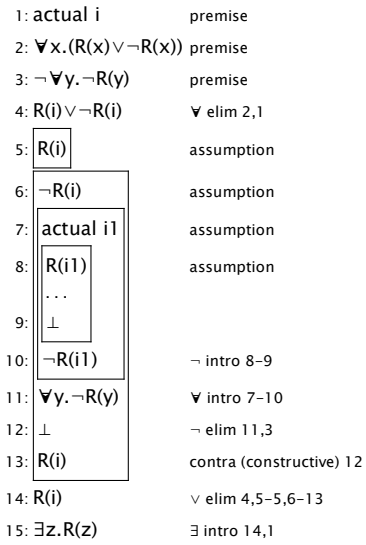

Fig. 16. A constructive failure

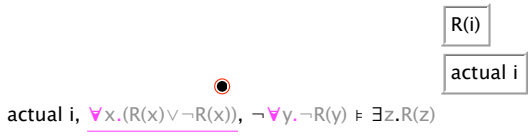

Fig. 17. The start of a constructive disproof

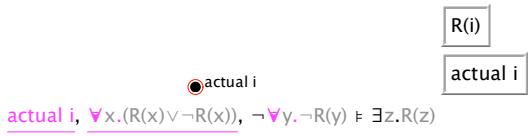

Fig. 18. An individual added

true in this case, because $R(x)$ and $R(y)$ aren't interesting in a universe containing an individual i: only $R(i)$ matters. I shall return to the issue of evaluation and colouring of quantified formulae below.
We can press on. The stuck proof suggests (because of its box structure) that we should make a new world containing an individual $i1$. It's easy to make a new world by option-dragging the root world, but we don't have a tile with actual $i1$. Jape lets us add individuals to the model by double-clicking an 'actual' tile. This solves our immediate problem, and we can drag and drop the result onto the upper world (figure 19).

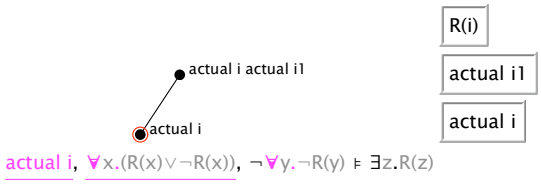

Fig. 19. A new world and a new individual

We still don't have $R(i1)$ to drop onto the model, but double-clicking a predicate tile gets you a novel instance built up from the available individuals (if there is more than one possibility, a choice dialogue lets you pick the one you want). That new instance can be dropped onto the diagram and the disproof is complete (figure 20) – all premises forced (shown by underlining) and the conclusion unforced.

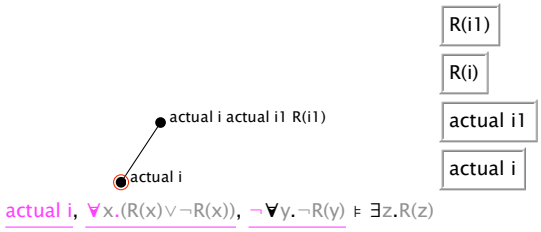

Fig. 20. A completed disproof

This is all very well: the stuck proof shows the expert how to proceed, and the interface supports all the necessary moves. But the display doesn't help the novice to understand *why* it's colouring the assertion as it does. Most of the syntax colouring is grey, telling us that the literal subformulae aren't used, and that there is more behind the scenes to be explained. Exploring the worlds doesn't help much at all, as figure 21 shows.
At this point I have to say that, as a designer, I'm stumped. I'd like to design an interaction that allows a novice to double-click (say) on $\exists z.R(z)$ and find out why it isn't forced at the base world (there's only

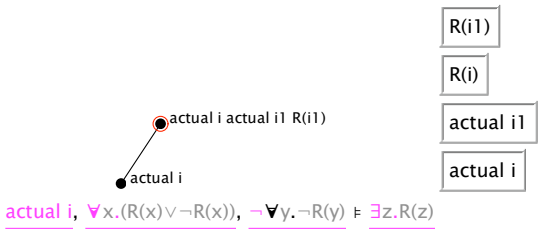

Fig. 21. A lot of colour and little illumination

one individual available and $R(i)$ isn't forced at the base world) or why conversely it is forced in the upper world (there we can call on $i1$ and $R(i1)$). That might be supported, in this trivial example, by a little drop-down menu, and the results might be helpful. But in a more complex example using nested quantifiers we would want to explore further, and need to explore what we see in a way that drop-down menus in the assertion don't seem to support.

6 Deficiencies

Explanation is partly supported in the propositional case, hardly supported at all in the predicate case, as discussed above. I suspect this is a considerable stumbling block for the average novice.

Layout of the labels decorating a world is primitive: they are shown as a single line of text extending right from the world. In a large diagram those text lines confusingly overlap other objects.

When drawing complex structures a single-history undo is sometimes irksome. It would be worth experimenting with an object-focussed undo, though I'm conscious of the need to preserve a quiet interface.

7 Evaluation

This mechanism hasn't been independently evaluated. It's been given to two successive first-year classes of undergraduates to use, in the first year under my supervision. My informal assessment is that students find Jape's proof mechanisms, especially since they were subjected to thoughtful evaluation in [2], relatively easy to use, and formal proof correspondingly relatively easy to learn. They find disproof harder to understand and in practice they struggle with the disproof interface. (Then most of them then find Hoare logic, which was taught without Jape support in the final section of the course, quite impenetrable, but that's another story!)

On the other hand many of them do learn about models and disproof using Jape, and all appear to be grateful for the unbiased verdict of a calculator on their attempts to answer exercises. The interaction with the disproof calculator is much more like what many of them would hope to

have with a proof calculator: they can put in a copy of what they have written on paper, and receive a judgement, provided that their pencil and paper version is not too outlandish.
Clearly it would need a dispassionate evaluation by a trained educationalist to tease out the interface difficulties from the conceptual misunderstandings, so that interactive disproof in Jape can perhaps become as smoothly successful as its proof mechanisms.

Acknowledgements

Bernard Sufrin and I designed and implemented Jape together for several years. His architectural insight, and crucial early design decisions, have been essential underpinning for everything I have managed to achieve since he moved on to other things. His continued advice helps me implement along the narrow path.
David Pym persuaded me to take Kripke semantics seriously as an educational target. The hundred and eighty students who suffered the first time I tried to teach it without mechanical support provided the evidence which underpinned the case for a disproof calculator.
Paul Taylor, Jules Bean, Mike Samuels and Graham White (who invented the 'scrabble tile' layout) advised me and supported me whilst I built the disproof parts of Jape. Paul's ideas on explanation were particularly influential.
Even more than usual, mistakes in Jape are all my own fault.

References

1. The jape web site. http://www.jape.org.uk. Latest versions of Jape for various platforms, including the encoding of natural deduction discussed in this paper.
2. J. C. Aczel, P. Fung, R. Bornat, M. Oliver, T. OShea, and B. Sufrin. Using computers to learn logic: undergraduates experiences. In G. Cumming, T. Okamoto, and L. Gomez, editors, *Advanced Research in Computers and Communications in Education: Proceedings of the 7th International Conference on Computers in Education*, Amsterdam, 1999. IOS Press.
3. R. Bornat and B. Sufrin. Jape: A calculator for animating proof-on-paper. In William McCune, editor, *Proceedings of the 14th International Conference on Automated deduction*, volume 1249 of *LNAI*, pages 412–415, Berlin, July 13–17 1997. Springer.
4. Richard Bornat. Rendering tree proofs in box style. submitted to UITP 03, 2003.
5. Richard Bornat and Bernard Sufrin. Animating formal proof at the surface: The Jape proof calculator. *The Computer Journal*, 42(3):177–192, 1999.
6. Richard Bornat and Bernard Sufrin. A minimal graphical user interface for the jape proof calculator. *Formal Aspects of Computing*, 11(3):244–271, 1999.

7. M. Dummett. *Elements of Intuitionism*. Oxford Logic Guides. Clarendon Press, Oxford, 1977.
8. S. A. Kripke. Semantical analysis of intuitionistic logic. In J. Crossley and M. A. E. Dummett, editors, *Formal Systems and Recursive Functions*, pages 92–130. North-Holland, Amsterdam, 1965.