Disproof in Jape (a short how-to note)

Richard Bornat (richard@bornat.me.uk)

December 15, 2003

The `examples/natural_deduction/I2L.jt` encoding of natural deduction in Jape allows for *disproof* as well as proof. It allows you to make Kripke diagrams which Jape will check to see if they are examples or counter-examples (or neither) against a sequent which you choose.

If you have no notion what a Kripke diagram is, then maybe you can find out by playing, or maybe you will have to wait for my book (which will come out, so long as I deliver it to OUP in time!). This note is just about the mechanics, and what Jape can show you.

This note shows coloured diagrams, and may not make sense if printed in shades of grey.

1 Basics

At any point during a proof attempt you can invoke 'Disprove' from the Edit menu.[1] Jape splits the window and shows you a diagram with a single empty world and, by default, the proof-attempt sequent (see below for how to get a different sequent).

Figure 1 shows an example of what you see. The main things to notice are

- the single world is ringed in red;

- some of the names and connectives in the sequent are violet, some are grey, and some are black;

- some of the formulae are underlined and some are not;

- the sequent uses the semantic turnstile ⊨;

- there's a little green waste bin on the right;

- above the waste bin there are some tiles, each containing an atomic formula.

---

[1] There isn't a Disprove button on the Conjectures menus. Perhaps there should be! At present you have to hit Prove to start a prove attempt and then choose Disprove.
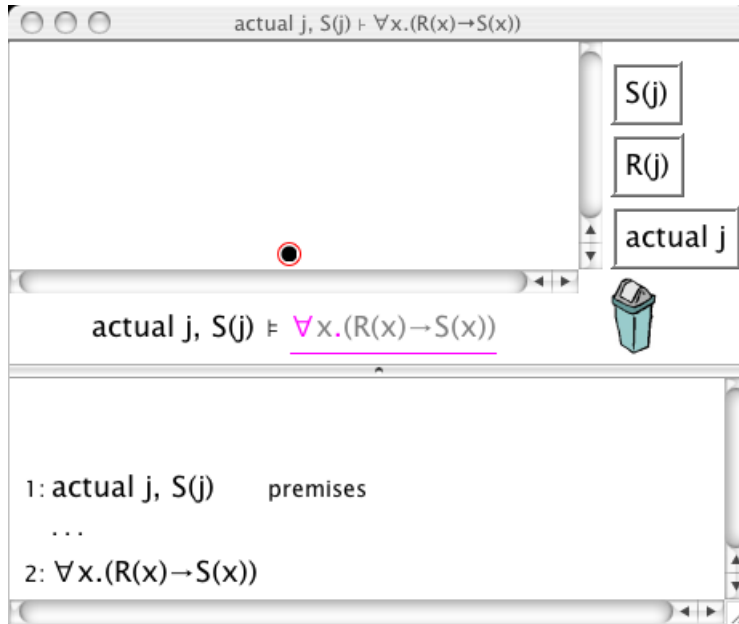
Figure 1: The start of a disproof attempt

Your window may not look exactly like figure 1, but it will have the same parts. Other illustrations in this document show the important bits of the proof and/or disproof sections of the window, without the waste bin and the scroll bars and stuff.

1.1   Alternative sequents

If, during a proof attempt, you select a conclusion (or, if Jape doesn't want to let you do that, the reason next to the conclusion) and then hit Edit:Disprove, Jape will use the disproof sequent consisting of the conclusion formula and all the lines above it as hypothesis formulae. If you select some of the hypothesis lines, you can choose the collection of hypothesis formulae that appear.

You can choose a new disproof sequent at any time, even in the middle of a disproof attempt.

If you want to set up an entirely new challenge, enter it into one of the conjectures panels (New button), hit Prove and then Edit:Disprove in the proof window.

1.2   Selecting a situation

By selecting a world you define a 'situation' comprising the world and all the worlds you can reach above it. The root world of the situation is ringed in red to show that it is selected. There is always exactly one selected world, and therefore exactly one situation. Figure 2 shows a situation with more than one world; figure 3 shows a single-world situation inside a multi-world diagram.
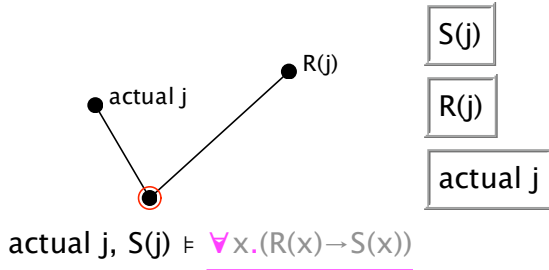
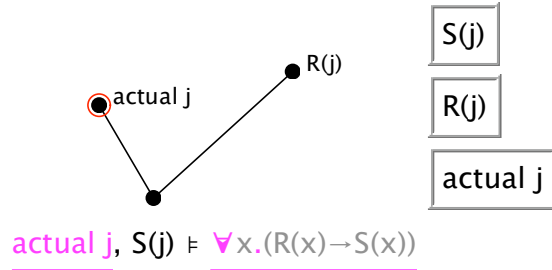Figure 2: A multi-world situation
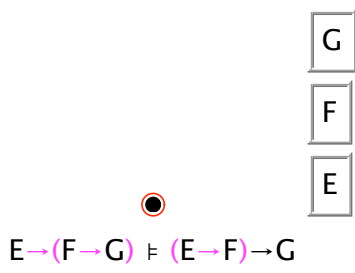

Figure 3: A single-world situation

Figure 4: A simple counter-example

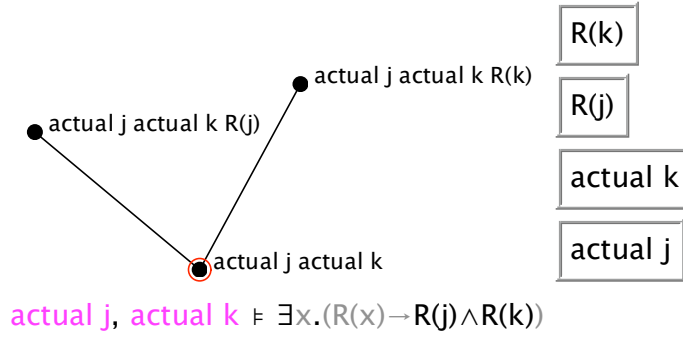

Figure 5: An elaborate counter-example

1.3 What's forced ? – colouring, greying, underlining

Jape evaluates all the formulae and sub-formulae in the sequent and colours them.

- The default colour is black, and it means *not forced*.

- If a (sub-)formula's value is never called on – usually this is true of quantified predicates, for example – then it's grey, meaning *strictly irrrelevant*.

- An atomic (sub-)formula which is *forced* is coloured violet (this covers atoms like E, atomic predicates like $S(j)$ and individuals like $\text{actual}\,k$).

- When a non-atomic (sub-)formula is forced, its connective is coloured violet.

- When an entire sequent hypothesis or conclusion is forced, it is underlined in violet.

So when all the premises are underlined and the conclusion is not, you have a counter-example; when everything is underlined you have an example; otherwise you have nothing in particular.

Figure 4 shows a sample counter-example, and the colouring gives a hint of why that is: E isn't forced anywhere, so the hypothesis implication is forced; and then, because $E \rightarrow F$ is forced but G isn't, the conclusion isn't forced. Figure 5 shows an elaborate counter-example, and the colouring isn't much help at all.[2]

---

[2] I have some ideas which might help in this case, but I haven't implemented them yet.

3

Figure 1 is neither a counter-example nor an example. The hypotheses aren't forced because, as atomic formulae, they aren't present at the root world; the conclusion is forced because every individual in the situation (there aren't any ..) forces a version of $R(x) \to S(x)$. Similar remarks apply to figures 2 and 3.

2 Making diagrams

To begin your disproof Jape presents you with the simplest diagram: the isolated empty world. You add parts of your diagram – worlds, formulae and lines – using drag-and-drop mouse gestures. You can delete the same components by dragging them to the waste bin. You can add tiles with double-clicks. You can Undo your actions to any degree that you like, and Redo likewise.[3]

2.1 Making new worlds

By dragging on a world with the middle mouse button or by holding down the alt shift key while dragging with a single-button mouse, you drag a copy of the world with a line attaching it to the world it came from.

The new world stays where you put it. If it's above the world you dragged it from, they will be connected by a line. If it's level or below, then no line.

Jape preserves monotonicity, so the new world will contain all the formulae that the original world did.

2.2 Dragging worlds

You can drag worlds about the place to make your diagram look nicer. You use the left mouse button on a multibutton mouse. Jape will delete a line if you drag a world below its parent.

You can drag a world onto a line (which lights up to show you are over it) or another world (which lights up ditto) and Jape does the obvious thing, adding whatever formulae are necessary to whatever worlds in order to maintain monotonicity.

If you drag a world to the waste bin it's deleted, along with all the formulae attached to it and all the lines connected to it. The bin lights up when it will accept the world; it won't accept the currently-selected world.

2.3 Making lines

If you want to make a line between world A and world B, where A is below B, make-world-drag from A to B and drop the new world onto B. Jape makes the monotonicity adjustments. Dragging from B to A has some effect, but perhaps not a very useful effect (but there's always Undo!).

---

[3]Undo and Redo apply to the last pane you clicked, either proof or disproof.

2.4 Dragging lines

You can drag lines and drop them onto worlds or into the waste bin, as you wish. Jape does the obvious thing in each case (if you can't guess what that is, just try it!).

2.5 Dragging formulae

You can drag a formula from a tile and drop it onto a world (the world lights up, of course, to show that it's eager to accept; worlds won't accept a second copy of a formula they already have). The formula isn't deleted from the tile.

You can drag a formula from one world onto another; it is added to the destination world (and not deleted from the source world).

Jape preserves monotonicity, so all worlds reachable from the destination world get a copy of the dropped formula as well.

If you let go of a formula when you aren't over a world that has lit up to accept it, it flies back to base.

If you drag a formula from a world to the waste bin it is deleted from that world (and then, to preserve monotonicity, from its parents and its grandparents and so on down).

3 Making individuals and predicate instances

If a sequent mentions an individual by including a hypothesis like $\text{actual}\,j$, then there will be a tile for that individual (see figures 1, 2, 3 and 5, for example). If there are quantifiers but no individuals you will get a free $\text{actual}\,i$ tile.

If you need more individuals, double-click any individual tile. Jape uses a boring but obvious numbering algorithm to name the new individual – $\text{actual}\,j$, for example, generates first $\text{actual}\,j1$ then $\text{actual}\,j2$, and so on.

If you want to make a new predicate instance, double-click a predicate-instance tile. Jape looks through your tiles for unused individuals and uses them to give you a new one. For example if you have a tile showing $S(j)$ and individuals $\text{actual}\,j$ and $\text{actual}\,j1$, then double-clicking $S(j)$ gets you $S(j1)$, as you might expect. If there's more than one instance that could be made, Jape shows you the alternatives and asks you to choose. If you don't have enough individuals to make a new instance, it tells you so.
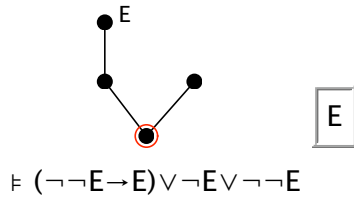
⊨ (¬¬E→E)∨¬E∨¬¬E

| E |

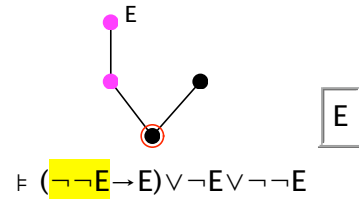Figure 6: A counter-example

⊨ (¬¬E→E)∨¬E∨¬¬E

| E |

Figure 7: ¬¬E forced here

⊨ (¬¬E→E)∨¬E∨¬¬E

| E |
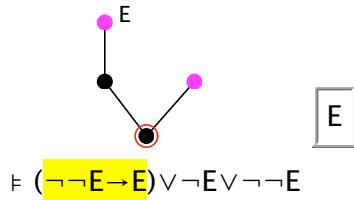
Figure 8: ¬¬$E \rightarrow E$ forced here

⊨ (¬¬E→E)∨¬E∨¬¬E

| E |
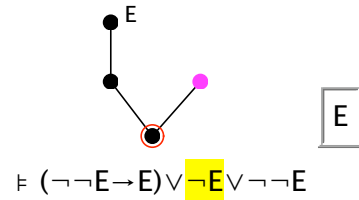
Figure 9: ¬E forced here
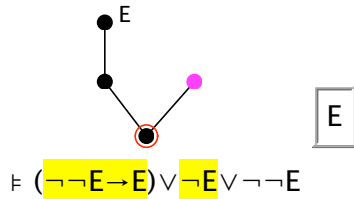
⊨ (¬¬E→E)∨¬E∨¬¬E

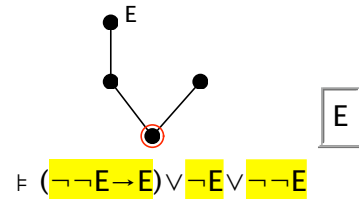| E |

Figure 10: Multiple forcing

⊨ (¬¬E→E)∨¬E∨¬¬E

| E |

Figure 11: Too much forcing

4   Exploring reasons

The hardest thing to explain to a novice is *why* a particular formula is forced in a particular situation. Figure 6 shows a particularly ludicrous formula and its counter-example, which really needs explanation.

Jape provides some mechanisms which can be some help (in versions v6.2.a1.2 and later).

First, the colouring of atomic (sub-)formulae and connectives can help. But with the negative connectives – $\rightarrow$ and ¬ – and with the quantifiers, we often need more help.

Second, Jape allows you to *text-select* – alt+press-and-drag, or middlebutton press-and-drag – a subformula (or select a formula) in the sequent. It will then colour violet all the worlds which force the selected (sub)formula. For example, figure 7 shows where ¬¬E is forced in our difficult example. Figures 8 and 9 show information about some other subformulae.

4.1   Multiple selections

If you choose more than one formula, as in figure 10, for example, Jape shows you where they are *all* forced. If you choose too many formulae, as in figure 11, you may find that there is no subsituation which forces
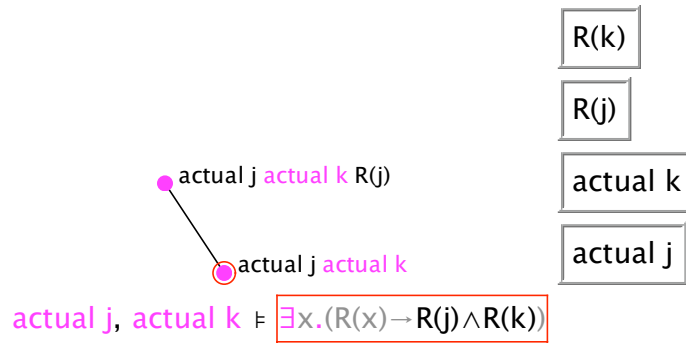
Figure 12: Support for a quantifier

them all. But that's explanation too!

4.2 Quantifier reasons

The world-colouring trick works well, but it isn't enough when you are dealing with quantifiers. You would like to know which individual formulae in the tree support the quantifier, and which do not. Jape can do that (in versions v6.2.a1.3 and later) by colouring individuals in the tree. In figure 12, for example, worlds which force the selected quantifier are violet; individuals which support the quantifier (those which generate a forced formula if you instantiate the quantified formula with them) are violet too. You can work out that $R(k) \rightarrow R(j) @ R(j)$ at each of the worlds of figure 12.

This isn't enough in general. You would like to be able to see those instantiations and play with them – see why they are forced (or not, as the case may be). I think I can see how to build a Jape which could do that, but it doesn't do it at present.

How to complain

If Jape doesn't work for you, it doesn't work. It's best to send your tales of woe to bugs@jape.org.uk. One day there'll be a mechanism to let you see other people's messages, but at the moment it's a private conversation.