

Proofs of some pointer programs

Part 1: Background

Richard Bornat, unattached, October 2002

These proofs are published as an awful warning: this way of proving pointer programs is a dead end. On the same basis that TV people call programs about extreme weather “weather porn”, this is proof porn. Watch me crash and burn. Enjoy.

When I did these proofs I was very proud. They were the largest collection of proofs yet done in Jape (still are, so far as I know). Nobody, so far as I knew, had done entirely mechanical proofs of such large pointer-manipulating programs¹. This was hubris, of course, and my self-satisfaction was soon undermined. O’Hearn, Reynolds and others, partly prompted by my success but prompted even more by the evident difficulty of attempts like those reported here, produced a separation logic that made much of my efforts unnecessary.

In making these proofs I spent most of my time proving separation. It was necessary to establish and to re-establish, over and over again, that changing a tiny bit of a cell one data structure, which was specified separate in memory from a second data structure, kept the structures separate. Nowadays that usually emerges naturally from use of the operator ‘*’ in separation logic, and that fact alone means that all my striving was futile.

This document details four large groups of proofs: background theorems, list reversal, list merge and Schorr-Waite, all described in the MPC paper of 2000. The proofs are shown in the order they appear in the panels. Most of them are pretty trivial; their very existence proves that Jape is a calculator, not a theorem prover.

If you want to see the calculator in action, you have to try the proofs for yourself (currently, only MacOS Classic has an appropriate implementation of Jape).

0.1 Apology

My humble, genuine and heartfelt apologies to anybody who, prompted by my citation in the 2000 MPC paper, looked for and couldn’t find this material in the place that I said it would be. No deception was intended: the proofs did exist, but because of pressure of work (the cause, eventually, of my unattached status) I couldn’t find time to write this explanation and to generate the necessary pictures and text. When I did have time, I found that the font in which I’d developed them had been lost on the disc of a stolen laptop, and various improvements to Jape in the meantime had unfortunately made the saved proofs unloadable. I’ve fixed the problems, improved the presentation of a few of the proofs, and here they are at last.

1.1. Basic Japery

I built a great deal of Japeish code to allow me to prove things conveniently. You may wish to skip this section at first reading; you may want to read ‘Roll your own Jape’ in order to understand it. Every indented line which follows comes from the source files, but for ease of explanation I’ve rearranged the order in which they appear.

¹ I discovered when preparing this note that the Schorr-Waite proof is incomplete: I didn’t develop a measure for the loop, so no proof of termination.

1.1.1. Remark on forward reasoning

For a long time Jape has had a treatment of cut rules which facilitated ‘forward reasoning’ – taking a hypothesis and generating more hypotheses from it – in a box-and-line proof. The cut rule is normally²

FROM $\Gamma \vdash B$ AND $\Gamma, B \vdash C$ INFER $\Gamma \vdash C$

The basic mechanism is that cut steps are hidden in the displayed proof, and the conclusion B of the left antecedent, once proved, is available as a hypothesis (i.e. just like an assumption) in the derivation of C .

This mechanism has been taken a good deal forward in the version of Jape used in these proofs and used in teaching at Queen Mary College since 2000 (a version is available for Linux and MacOS Classic but hasn’t been widely distributed; it will appear publicly as soon as I’ve finished the tri-platform GUI implementation, which is only delayed by the writing of this note :-)). Now it is possible to insert cuts anywhere in the tree (CUTIN) and the unproved consequent of a cut can be selected either as a conclusion (for proving) or as a hypothesis (for later proofs).

What CUTIN tac does is to look down the tree for the lowest point which has the same left context (hypotheses) as the current goal, to insert a cut there, and to apply tac to its left antecedent B . Then B can be used as a hypothesis in the tree above the new cut: the effect is that, at last, it is possible to construct box-and-line proofs in which every preceding reachable line can be used as a hypothesis.³

1.1.2. Remark on abbreviations

In order to be able to read the proofs, it’s helpful to be able to abbreviate formulæ: writing P , for example, in place of a long invariant formula. Jape ought to have a notation of ‘defined equivalence’ to help in such a case, but it doesn’t (yet).

So I employed a trick: I used constant names like PP and QQ to describe theorems, making rules like

RULE "QQ $\hat{=}$..." IS QQ $\hat{=}$ list($p \Rightarrow t \Rightarrow \text{nil}$) \wedge $p \Rightarrow t \Rightarrow \text{nil} = S$

I put those rules in a Definitions panel (see below).

It’s a trick, and it means that the proofs are less mechanical than they ought to be. Substituting t' for t in QQ has no effect (it’s a constant name), but the same substitution applied to list($p \Rightarrow t \Rightarrow \text{nil}$) \wedge $p \Rightarrow t \Rightarrow \text{nil} = S$ has a good deal of effect. So it’s necessary to eliminate the pseudo-constants before they confuse Jape.

1.1.3. Remark on the presentations.

Each little proof, of a lemma or a major theorem, is given in three parts. First a statement of the problem in Japeish

DERIVED RULE IS FROM $A \vee B$ AND $\neg A$ INFER B

² Jape can deal with other versions of cut, but that’s irrelevant here.

³ But it isn’t easy, as you will find if you experiment with the logic used in these proofs: I only achieved patchy status. Nevertheless, I’m extremely proud of the programming that made it possible to record a path in the tree (LETGOALPATH), make a cut in the middle of the tree, and the path (GOALPATH) still takes you back to where you started from. It’s all pointless in the end, though, because I now realise that Jape should be a natural deduction box-and-line engine and not use proof trees at all ...

then the proof as it is displayed on a Jape screen

```

1:  $A \vee B$   $A \vee B$ 
2:  $A$  assumption
3:  $\neg A$   $\neg A$ 
4:  $\perp$   $\neg\text{-E } 2,3$ 
5:  $B$   $\perp\text{-E } 4$ 
6:  $B$  assumption
7:  $B$   $\vee\text{-E } 1,2\text{-}5,6\text{-}6$ 

```

Given:

$A \vee B$

$\neg A$

(note that in making the proofs I've made decisions to hide some steps which I've felt to be irrelevant)

and finally a machine-readable version of the proof as it's output by Jape's Save Proofs command (with elision of some unnecessary enclosing material).

FORMULAE 0 B, 1 A, 2 $A \vee B$

SEQ

```

(cut«2,0/B,C»)
(GIVEN 0)
("∨-E"«1,0,0/A,B,C»)
(hyp«2/A»)
("⊥-E"«0/A»)
("¬-E"«1/B»)
(hyp«1/A»)
(GIVEN 1)
(hyp«0/A»)

```

The FORMULAE section is a table of formulae, included because in large proofs it saves a lot of time; the tactic which follows refers to elements of the table by number. At the time of writing Jape doesn't always include a record of the context in the machine-readable proof, which reduces their volume by an order of magnitude, but means that you only glimpse the complexity of what is going on. I hope you are thankful for being relieved of that burden.

In an ideal world I would have included only the statement and the proof display, and directed my readers to the machine-readable text on the web, or invited them to run Jape for themselves and duplicate my efforts. At the time of writing, however, Jape's implementations and fonts are not all they ought to be, so I've included the machine-readable version as well, bloating this commentary many times more than somewhat. I thought it worthwhile in case anybody wants to see the insides as well as the presentation.

1.1.4. Global variables

```

INITIALISE displaystyle box
INITIALISE autoAdditiveLeft true /* avoid explicit statement of left context */
INITIALISE hidetransitivity true
INITIALISE hidereflexivity true
INITIALISE tryresolution false

```

Displaystyle box means show me box and line proofs. AutoadditiveLeft true means assume left context is carried through inference steps: for example,

```
RULE "∨-E"(A,B) IS FROM  $A \vee B$  AND  $A \vdash C$  AND  $B \vdash C$  INFER C
```

is read as if it had been written

```
RULE "∨-E"(A,B,Γ) IS FROM  $\Gamma \vdash A \vee B$  AND  $\Gamma, A \vdash C$  AND  $\Gamma, B \vdash C$  INFER  $\Gamma \vdash C$ 
```

Hidetransitivity and hidereflexivity true activate a feature of Jape designed to present sequences of steps like $A=B, B=C, C=D, \dots, Y=Z$ as $A, =B, =C, =D, \dots, =Y, =Z$. As part of the mechanism it's useful to treat reflexive steps $K=K$ as we treat identity steps in box-and-line proofs: i.e. to hide them. Transitivity rules (antecedents $\Gamma \vdash X <op> Y$ and $\Gamma \vdash Y=Z$, consequent $\Gamma \vdash X=Z$) and reflexivity rules (no antecedents, consequent of the form $\Gamma \vdash X <op> X$) must be declared -- see below.

Tryresolution false turns off an ancient feature of Jape which, IMHO, shouldn't be there at all.

1.1.5. Syntax

The syntax is scattered throughout the various theories. For simplicity I've gathered it all together here.

1.1.5.1. Identifier syntax

CLASS ... means Jape will generate any number of versions of the name, if it appears in a rule or a theorem.

```
CLASS CONSTANT a b c d e f v w
CLASS NUMBER n i j
CLASS VARIABLE x y z
CLASS FORMULA A B C D E F G H I J P Q R S T V X Y
```

Particular names which mean something to us

```
CONSTANT nil true false ⊥
```

Names of program variables (Jape really *needs* a way to allow the user to extend the number of syntactic categories ...)

```
CONSTANT p q r t0
```

Names of defined functions and predicates (see below for definitions).

```
CONSTANT list rev perm length finitesequence hd tl oseq olist
CONSTANT U set finiteset finiteheight
```

The constant instruction of Hoare logic

```
CONSTANT skip
```

The various pseudo-constant formula names.

```
CONSTANT PP QQ RR TT SS
```

1.1.5.2. Sequent syntax

```
SEQUENT IS BAG ⊢ FORMULA
```

The context is an unordered collection of formulæ with repetitions allowed; the consequent is a single formula. That is, just as in natural deduction.

1.1.5.3. Operators and brackets

Jape allows users to define the syntax of formulæ using numerical precedences.

Substitution $E \ll F/x \gg$ and juxtaposition $E F$ have the highest priority. I define the appearance of substitution formulæ.

```
SUBSTFIX 10000 « E / x » /* so that { }, [ ] are available for other uses */
JUXTFIX 9000
```

Dot suffixing is pretty high precedence, and binds to the left.

```
INFIX 1000L .
```

The operators which describe sequences of locations in the heap (see below).

```
INFIX 450L ⇒ +⇒ ⇒+ ⇐ +⇐ ⇐+
```

Functions defined as operators.

```
INFIX 400L @ ||| ∪
```

INFIX 300L $\neg \cap \in \neg \in \subseteq$
 INFIX 250L \mapsto
 INFIX 200L \oplus

Logical, relational and arithmetic operators

PREFIX 350 \neg
 INFIX 300L $< > \leq \geq \neq = \equiv \neg \equiv$

Brackets don't have precedences. But unclosed opening brackets (LEFTFIX) and unopened closing brackets (RIGHTFIX) need one. These two define constructs of the form $\forall A:B$ and $\exists A:B$.

LEFTFIX 180 $\forall :$
 LEFTFIX 180 $\exists :$

More logical, relational and arithmetic operators.

INFIX 140L $\wedge \wedge \wedge$
 INFIX 120L $\vee \vee \vee$
 INFIX 100R $\rightarrow \leftrightarrow$

The operators of Hoare logic.

INFIX 60L $:=$
 INFIX 55L $;$

Lowest priority is the 'equal by definition' operator.

INFIX 50L \doteq

A variety of brackets, for assertions, indices and sequences (normal parentheses are built-in to Jape, as is comma).

OUTFIX { }
 OUTFIX []
 OUTFIX ()

More elaborate brackets for program logic.

OUTFIX if then else fi
 OUTFIX while do od

Brackets for sequence and set comprehensions.

OUTFIX (|)
 OUTFIX (| |)
 OUTFIX { | }
 OUTFIX { | | }

1.1.5.4. Binders

Jape doesn't automatically recognise binding constructs, but they can be declared. Unfortunately you have to declare all the variant forms that you use.

BIND x SCOPE P IN $\forall x : P$
 BIND x y SCOPE P IN $\forall(x,y) : P$
 BIND x SCOPE P IN $\exists x : P$
 BIND x y SCOPE P IN $\exists (x,y) : P$

The binding structure of sequence and set comprehensions.

BIND x SCOPE A IN (A | $x \in B$)
 BIND x SCOPE A C IN (A | $x \in B$ | C)
 BIND x SCOPE A IN { A | $x \in B$ }
 BIND x SCOPE A C IN { A | $x \in B$ | C }

1.1.6. Rules

Jape makes proofs by making rule-steps. It doesn't question rules, but neither will it go outside the rules to make a proof. Of course it's possible to cheat (see 'assert' below), and of course it's possible to go wrong and define an inconsistent system of rules. So the rules we use should be carefully scrutinised. Not all the rules presented in this document are used in proofs.

1.1.6.1. Rewriting

At the root of Jape's mechanisms is substitution. Substitution of equals is at the heart of various proofs.

```
RULE "rewrite="(A, OBJECT xx) IS FROM A=B AND P«B/xx» INFER P«A/xx»
RULE "rewrite1="(A, OBJECT xx) IS FROM A=B INFER P«A/xx»=P«B/xx»
RULE "rewrite≐="(A, OBJECT xx) IS FROM A≐B AND P«B/xx» INFER P«A/xx»
```

1.1.6.2. Equality and ordering

Rules can be presented without a name, in which case a string version of the consequent of the rule becomes its name. I am treating equality classically, and I'm still worried about that. It should be treated by 'definedness' considerations: for example (see below) if you know if A then B else C fi has a value, then certainly $A \vee \neg A$. But it's perhaps too late to be worrying about tidying all this up.

```
RULE IS  $\neg(A=B) \doteq A \neq B$ 
RULE IS  $\neg(A \neq B) \doteq A=B$  /* classical, classical? */

RULE IS  $A \leq B \doteq B \geq A$ 
RULE IS  $A \leq B \doteq A < B \vee A=B$ 
RULE IS  $A < B \doteq B > A$ 
RULE IS  $\neg(A < B) \doteq A \geq B$ 
RULE IS  $\neg(A \leq B) \doteq A > B$ 

RULE IS  $A=B \vee \neg(A=B)$  /* this is in preparation for definedness */
```

I'm not sure why these next two are here. I think it was preparation for a treatment of definedness ... I don't think I use them.

```
RULE IS  $P \doteq P=true$ 
RULE IS  $\neg P \doteq \neg(P=true)$ 
```

Transitivity, reflexivity and symmetry are exploited a lot.

```
RULE "transitive=" IS FROM A=B AND B=C INFER A=C
RULE "reflexive=" IS INFER A=A
RULE "symmetric=" IS FROM A=B INFER B=A
RULE "symmetric≠" IS FROM A≠B INFER B≠A

RULE "symmetric≐" IS FROM A≐B INFER B≐A
```

We tell Jape about the properties of some of the rules. This enables it to simplify the presentation of proofs (see above).

```
TRANSITIVE "transitive="
REFLEXIVE "reflexive="
```

1.1.6.3. Conditions

```
RULE IS FROM A INFER if A then B else C fi  $\doteq B$ 
RULE IS FROM  $\neg A$  INFER if A then B else C fi  $\doteq C$ 
```

If you can evaluate a conditional at all, then its test formula has a value. (This is the first and best sight of definedness in my treatment.)

```
RULE IS FROM if A then B else C fi INFER  $A \vee \neg A$ 
RULE IS FROM if A then B else C fi = D INFER  $A \vee \neg A$ 
```

1.1.7. Tactics

Even I don't recall how all this works. Jape will only construct proofs made out of rules, so no matter how bizarre the tactics employed, if the system of rules is sound, the proofs will be reliable.

1.1.7.1. Conveniences

```
TACTIC Fail(x) IS (SEQ (ALERT x) STOP)
```

1.1.7.2. Forward steps

The basic technique: remember where you are, make the step, choose the nth subgoal of the result, apply hyp to it (because we are making a forward step from the corresponding hypothesis), return to the position you started from and choose the next open goal.

```
TACTIC Forward(n,step)
  (LETGOALPATH G (WITHARGSEL step)
   (GOALPATH (SUBGOAL G n))
   (WITHHYPSEL hyp)
   (GOALPATH G)
   NEXTGOAL)
```

Make a Forward move, but start with a cut and make the move on its first antecedent (B in the cut rule).

```
TACTIC ForwardCut (n,step) IS (CUTIN (Forward n step))
```

If a hypothesis is selected, try to make a forward step, otherwise make a normal step.

```
TACTIC ForwardOrBackward (forward,step, stepname) IS
  WHEN
    (LETHYP _P
     (ALT (forward step)
          (WHEN
            (LETARGSEL _Q
              (Fail ("%s is not applicable to assumption %t with argument %t",
                    stepname, _P, _Q)))
              (Fail ("%s is not applicable to assumption %t", stepname, _P))))))
    (ALT (WITHSELECTIONS step)
         (WHEN
           (LETARGSEL _Q (Fail ("%s is not applicable with argument %t",
                               stepname, _Q)))
           (Fail ("%s is not applicable", stepname))))))
```

To apply a theorem forward.

```
TACTIC thmfwd(t) IS
  CUTIN
    (WHEN (LETHYP _P (ALT (WITHHYPSEL t) (SEQ t (WITHHYPSEL hyp))))
          t)
```

To make a 'true forward' step, and to allow the results to be hypotheses in later steps.

```
MACRO trueforward(tac) IS
  (LETGOAL _A (CUTIN (LETGOAL _B (UNIFY _A _B) tac)) (MATCH (ANY hyp)))

TACTIC trueforwardGOALPATH IS
  trueforward (QUOTE (LETGOALPATH G (ASSIGN tacticresult G)))
```

More support for forward steps.

```
TACTIC Forwardspecial (action, stepname) IS
  WHEN
    (LETHYP _P action)
    (Fail ("%s should be used with a hypothesis selected", stepname))

TACTIC Forwardspecialcut(n,step,stepname) IS
  ForwardOrBackward(Forwardcut n) step stepname
```

(There's also a tactic called ForwardSubstHiding, but it's not used anywhere, it's complicated and I don't understand it :-)) so I've omitted it.)

1.1.7.3. rewriting

Rewriting gets quite a bit of support. Basically, if you select a subformula then Jape can notice the fact with various LET...SUBSTSEL tacticals, it can treat your selection as defining a substitution, and it can use that substitution to do things to the proof. Substitutions can be applied to rules, replacing parameters with arguments.

This stuff is complicated: I'll explain it another day.

```
TACTIC byrewrite (rew) IS
  WHEN
    (LETCONCSUBSTSEL (_P«_A/_xx») rew«_P,_A,_xx/P,A,xx»)
    (LETHYPSUBSTSEL (_P«_A/_xx»)
      (CUTIN
        (LETGOALPATH G rew (GOALPATH (SUBGOAL G 1)) (WITHSUBSTSEL hyp)
          (GOALPATH G) NEXTGOAL)))
    (Fail "no selection")
TACTIC "byrewrite*" (rew) IS
  WHEN
    (LETCONCSUBSTSEL (_P«_A/_xx») rew«_A/A»)
    (LETHYPSUBSTSEL (_P«_A/_xx»)
      (CUTIN
        (LETGOALPATH G rew«_A/B» (GOALPATH (SUBGOAL G 1)) (WITHHYPSSEL hyp)
          (GOALPATH G) NEXTGOAL)))
    (LETARGSEL _A
      (WHEN
        (LETHYP _P
          (CUTIN
            (LETGOALPATH G
              rew«_A/B»(GOALPATH (SUBGOAL G 1)) (WITHSUBSTSEL hyp)
              (GOALPATH G) NEXTGOAL)))
          rew«_A/A»))
    (Fail "no selection")
TACTIC evaltac (step, stepname) IS
  WHEN (LETHYPSUBSTSEL (_P«_A/_xx»)
    (CUTIN
      (LAYOUT stepname) /* lhs named for step */
      (LETGOALPATH G
        "rewrite="
        (LAYOUT HIDEROOT "symmetric="«_A/A»)
        step
        (GOALPATH (SUBGOAL G 1))
        (WITHSUBSTSEL hyp)
        (GOALPATH G)
        NEXTGOAL)))
    (LETCONCSUBSTSEL _P
      (LAYOUT stepname)
      (WITHSUBSTSEL "rewrite=")
      step)
    step
TACTIC "eval*" (step, stepname) IS
  WHEN
    (LETARGSEL _A
      (WHEN
        (LETHYP _P
          (CUTIN
            (LAYOUT stepname) /* lhs named for step */
            (LETGOALPATH G
```

```

                "rewrite="«_A/B»
                (LAYOUT HIDEROOT "symmetric=")
                step
                (GOALPATH (SUBGOAL G 1))
                (WITHHYPSEL hyp)
                NEXTGOAL)))
        (LETGOAL _P
         (LAYOUT stepname)
         "rewrite="«_A/A»
         step)))
    step
TACTIC rewritehypformula(stepname,rewrite,action,close) IS
  CUTIN
  (LAYOUT stepname)
  rewrite
  (LETGOALPATH G
   action
   (GOALPATH (RIGHT G))
   close)
TACTIC rewriteL2R (rewrite, rev,tac) IS
  WHEN
  (LETHYPSUBSTSEL _P
   (rewritehypformula tac rewrite
    (LAYOUT HIDEROOT rev (LAYOUT HIDEROOT tac))
    (WITHSUBSTSEL hyp)))
  (LETCONCSUBSTSEL _P (LAYOUT tac) (WITHSUBSTSEL rewrite) (LAYOUT HIDEROOT tac))
  (LETHYP _P
   (rewritehypformula tac rewrite
    (LAYOUT HIDEROOT rev«_P/A» (LAYOUT HIDEROOT tac))
    (WITHHYPSEL hyp)))
  (LETGOAL _P (LAYOUT tac) rewrite«_P/A» (LAYOUT HIDEROOT tac))
TACTIC rewriteR2L (rewrite, rev, tac) IS
  WHEN
  (LETHYPSUBSTSEL _P
   (rewritehypformula tac rewrite (LAYOUT HIDEROOT tac) (WITHSUBSTSEL hyp)))
  (LETCONCSUBSTSEL _P (LAYOUT tac) (WITHSUBSTSEL rewrite)
   (LAYOUT HIDEROOT rev) (LAYOUT HIDEROOT tac))
  (LETHYP _P (rewritehypformula tac (rewrite«_P/B»
   (LAYOUT HIDEROOT tac) (WITHHYPSEL hyp)))
  (LETGOAL _P (LAYOUT tac) rewrite (LAYOUT HIDEROOT rev«_P/B»)
   (LAYOUT HIDEROOT tac))
TACTIC symmetry(rule) IS
  WHEN
  (LETHYP _P
   (CUTIN (SEQ rule (WITHHYPSEL hyp))))
  (LAYOUT HIDEROOT rule)

```

1.1.7.4. iteration

```

TACTIC "DO+"(tac,fail) IS
  ALT (SEQ tac (DO tac))
  fail
TACTIC iterateL2R (rew,sym,pattern,rule,fail) IS
  WHEN
  (LETHYP _A (itforwardL2R rew sym _A pattern rule fail))
  ("DO+" (itstepL2R rew pattern rule) fail)
MACRO itstepL2R (rew,pattern, rule) IS
  LETGOAL _P
  (LETOCCURS pattern _P _Q

```

```

(LAYOUT HIDEROOT (LAYOUT "iterate" (1)))
rew«pattern/A»
rule)

TACTIC iterateR2L (rew,sym,pattern,rule,fail) IS
  WHEN
    (LETHYP _A (itforwardR2L rew _A pattern rule fail))
    ("DO+" (itstepR2L rew sym pattern rule) fail)

MACRO itstepR2L (rew, sym, pattern, rule) IS
  LETGOAL _P
  (LETOCCURS pattern _P _Q
  (LAYOUT HIDEROOT (LAYOUT "iterate" (1)))
  rew«pattern/A»
  sym
  rule)

MACRO itforwardstepR2L (rew, h, pattern, rule) IS
  (LETOCCURS pattern h _Q
  (LAYOUT rule (1))
  rew«pattern/B»
  rule
  (hyp h))

MACRO hideitforwardstep(pattern, G, A) IS
  WHEN
    (LETOCCURS pattern A _A1
    (GOALPATH (PARENT G)) (LAYOUT HIDE CUT))
  SKIP

TACTIC itforwardR2L (rew, h, pattern, rule, fail) IS
  ALT (SEQ (CUTIN
    (LETGOALPATH G
    (LETGOAL _A
    (itforwardstepR2L rew h pattern rule)
    (hideitforwardstep pattern G _A)
    (itforwardR2L rew _A pattern rule SKIP))))))
  fail

```

1.1.8. Menus and panels

Everything is controlled from menus and panels. Most of the panels include buttons like these, which enable you to use a definition to rewrite in either definition (depending on content, some use `rewrite=` rather than `rewrite≐`), to apply a theorem or to cut one into the proof:

```

BUTTON "A≐..." IS apply rewriteL2R "rewrite≐" "symmetric≐" COMMAND
BUTTON "...≐B" IS apply rewriteR2L "rewrite≐" "symmetric≐" COMMAND

BUTTON "Apply" IS apply COMMAND
BUTTON "Apply forward" IS apply thmfwd COMMAND

```

The Rewrite menu has some tactical mechanism in it:

```

ENTRY "reflexive=" IS (LAYOUT HIDEROOT "reflexive=")
ENTRY "symmetric=" IS symmetry "symmetric="
ENTRY "symmetric≠" IS symmetry "symmetric≠"
ENTRY "rewrite=" IS byrewrite "rewrite="
ENTRY "rewrite≐" IS byrewrite "rewrite≐"

```

1.2. Natural Deduction

1.2.1. Rules

1.2.1.1. Structural rules

```
RULE hyp(A) IS INFER A ⊢ A
RULE cut(B) IS FROM B AND B ⊢ C INFER C
RULE thin(A) IS FROM C INFER A ⊢ C
```

We tell Jape what these rules mean, which enables it to simplify the presentation of proofs.

```
IDENTITY    hyp
CUT         cut
WEAKEN     thin
```

Hyp steps are normally hidden (there's a global variable 'hidehyp' which controls this). This next makes Jape try hyp at the end of every step.

```
AUTOMATCH hyp
```

1.2.1.2. Connectives

Here %-I means intro, %-E means elimination. Apart from the misnaming of the rule \perp -E, which should have been called contradiction or something like that, I think this is an unexceptionable collection.

```
RULE "→-E"(A)      IS FROM A AND A→B INFER B
RULE "↔-E(L)"(B)   IS FROM A ↔ B INFER A→B
RULE "↔-E(R)"(A)   IS FROM A ↔ B INFER B→A
RULE "∧-E(L)"(B)   IS FROM A ∧ B INFER A
RULE "∧-E(R)"(A)   IS FROM A ∧ B INFER B
RULE "∨-E"(A,B)    IS FROM A ∨ B AND A ⊢ C AND B ⊢ C INFER C
RULE "¬-E"         IS FROM B AND ¬B INFER ⊥
RULE "⊥-E"(A)     IS FROM ⊥ INFER A
RULE "∀-E"(B)      IS FROM ∀x:A INFER A«B/x»
RULE "∃-E" (OBJECT cc) WHERE FRESH cc AND cc NOTIN ∃x:A
                   IS FROM ∃x:A AND A«cc/x» ⊢ C INFER C

RULE "→-I"        IS FROM A ⊢ B INFER A→B
RULE "↔-I"        IS FROM A→B AND B→A INFER A↔B
RULE "∧-I"        IS FROM A AND B INFER A ∧ B
RULE "∨-I(L)"(B)  IS FROM A INFER A ∨ B
RULE "∨-I(R)"(A)  IS FROM B INFER A ∨ B
RULE "¬-I"(A)     IS FROM A ⊢ ⊥ INFER ¬A
RULE "∀-I"(OBJECT cc) WHERE FRESH cc
                   IS FROM A«cc/x» INFER ∀x :A
RULE "∃-I"(B)     IS FROM A«B/x» INFER ∃x:A
```

1.2.1.3. C-style && and //

```
RULE IS A∧∧B ≐ if A then B else A fi
RULE IS A∨∨B ≐ if A then A else B fi
```

1.2.2. Tactics

Trickery to make it possible to make several \wedge -I or \wedge -E steps in one. LAYOUT COMPRESS hides the intermediate steps, shows only the leaves of the result. It uses a global 'tacticresult' to communicate. Ouch!

```
TACTIC "∧-I*" IS
  WHEN
    (LETGOAL (_P∧_Q)
      (LAYOUT COMPRESS "∧-I")
      "∧-I"
      "∧-I*")
```

```

      (LETMATCH _G tacticresult "∧-I*" (ASSIGN tacticresult _G)) /* take leftmost goal */)
      trueforwardGOALPATH

```

```

TACTIC "∧-I total" IS
  WHEN (LETGOAL (_P∧_Q) "∧-I*")
        (Fail "that's not an ∧ formula")

```

Similar trickery for \wedge -E (but even more complicated).

```

TACTIC "∧-E*step"(P, rule,H) IS
  WHEN
    (LETMATCH (_P∧_Q) P
      (CUTIN
        (LETGOALPATH G (GOALPATH (PARENT G)) (LAYOUT HIDE CUT) (GOALPATH G))
        rule
        (LETGOAL _A (UNIFY _A H) hyp)))
    (CUTIN (LAYOUT "∧-E") rule (LETGOAL _A (UNIFY _A H) hyp)))

```

```

TACTIC "∧-E*(P) IS
  WHEN
    (LETMATCH (_P∧_Q) P
      ("∧-E*step" _P "∧-E(L)" P)
      ("∧-E*" _P)
      ("∧-E*step" _Q "∧-E(R)" P)
      ("∧-E*" _Q) )
    SKIP

```

```

TACTIC "∧-E total" IS
  WHEN (LETHYP (_P∧_Q) ("∧-E*" (_P∧_Q) ))
        (LETHYP _P (Fail "that's not an ∧ formula"))
        (Fail "no hypothesis selected")

```

```

TACTIC "Do ∧-E" (rule) IS
  WHEN (LETHYP _P (ForwardOrBackward (ForwardCut 0) rule "∧-E"))
        (ALT (SEQ (LAYOUT COMPRESS "∧-E") rule)
          (ForwardOrBackward (ForwardCut 0) rule "∧-E" ))

```

1.2.3. Menus and panels

The Logic menu has quite a bit of mechanism.

```

MENU Logic IS
  ...
  ENTRY "↔-I" IS
    SEQ "↔-I"
      (LETGOALPATH G
        (trueforward SKIP)
        (GOALPATH (RIGHT G)) (trueforward SKIP)
        (GOALPATH G))
  ENTRY "∧-I" IS
    SEQ (MATCH "∧-I total") (GOALPATH tacticresult)
  ENTRY "∨-I(L)" IS
    WHEN (LETGOAL _P (ForwardOrBackward (ForwardCut 0) "∨-I(L)" "∨-I"))
          (Forwardspecial (Forwardspecialcut 0 "∨-I(L)" "∨-I"))
  ENTRY "∨-I(R)" IS
    WHEN (LETGOAL _P (ForwardOrBackward (ForwardCut 0) "∨-I(R)" "∨-I"))
          (Forwardspecial (Forwardspecialcut 0 "∨-I(R)" "∨-I"))
  ENTRY "¬-I"
  ENTRY "∀-I" IS WHEN (LETGOAL (∀_xx:_A) (WITHARGSEL "∀-I")) "∀-I"
  ENTRY "∃-I" IS "∃-I with side condition hidden"
  ENTRY "→-E" IS
    WHEN (LETGOAL _P (ForwardOrBackward (ForwardCut 1) "→-E" "→-E"))
          (Forwardspecial (Forwardspecialcut 1 "→-E" "→-E" ))
  ENTRY "∧-E" IS

```

```

    WHEN (LETGOAL _P (MATCH "^-E total")) (Forwardspecial (MATCH "^-E total") "^-E")
  ENTRY "^-E(L)" IS
    WHEN (LETGOAL _P ("Do ^-E" "^-E(L)"))
      (Forwardspecial ("Do ^-E" "^-E(L)") "^-E")
  ENTRY "^-E(R)" IS
    WHEN (LETGOAL _P ("Do ^-E" "^-E(R)"))
      (Forwardspecial ("Do ^-E" "^-E(L)") "^-E")
  ENTRY "v-E" IS ForwardOrBackward (Forward 0) "v-E" "v-E"
  ENTRY "v-E" IS "v-Etac"
  ENTRY "v-E" IS (ForwardOrBackward (Forward 0) "v-E" "v-E")
END

```

A comment in the code claims “The next tactic is needed because we can't give an application of a tactical as an argument. But we'd probably include it anyway, for clarity.” That's untrue nowadays, because we can now include applications as arguments. But here it is anyway.

```

TACTIC "v-Etac" IS
  WHEN
    (LETHYP (v _xx : _A) (CUTIN (WITHARGSEL"v-E") (WITHHYPSEL hyp)))
    (LETHYP _P (Fail "that's not a v formula"))
    (LETCONCSUBSTSEL _P (WITHSUBSTSEL "v-E")) /* quick hack */
    (WITHSELECTIONS "v-E")

TACTIC "v-I with side condition hidden" IS LAYOUT "v-I" (0) (WITHARGSEL "v-I")

```

1.2.4. Proofs

DERIVED RULE IS FROM $A \vee B$ AND $\neg A$ INFER B

```

1:  $A \vee B$   $A \vee B$ 
2:  $A$  assumption
3:  $\neg A$   $\neg A$ 
4:  $\perp$   $\neg E$  2,3
5:  $B$   $\perp-E$  4
6:  $B$  assumption
7:  $B$   $\vee-E$  1,2-5,6-6

```

Given:

$A \vee B$
 $\neg A$

FORMULAE 0 B, 1 A, 2 $A \vee B$

SEQ

```

(cut«2,0/B,C»)
(GIVEN 0)
("v-E"«1,0,0/A,B,C»)
(hyp«2/A»)
("⊥-E"«0/A»)
("¬-E"«1/B»)
(hyp«1/A»)
(GIVEN 1)
(hyp«0/A»)

```

DERIVED RULE IS FROM $A \vee B$ AND $\neg B$ INFER A

- 1: $A \vee B$ $A \vee B$
- 2: A assumption
- 3: B assumption
- 4: $\neg B$ $\neg B$
- 5: \perp $\neg E$ 3,4
- 6: A $\perp-E$ 5
- 7: A $\vee-E$ 1,2-2,3-6

Given:

$A \vee B$
 $\neg B$

FORMULAE 0 B , 1 A , 2 $A \vee B$

SEQ

(cut«2,1/B,C»)
 (GIVEN 0)
 ("∨-E"«1,0,1/A,B,C»)
 (hyp«2/A»)
 (hyp«1/A»)
 ("⊥-E"«1/A»)
 ("¬-E"«0/B»)
 (hyp«0/A»)
 (GIVEN 1)

DERIVED RULE IS FROM $\neg A \vee B$ AND A INFER B

- 1: $\neg A \vee B$ $\neg A \vee B$
- 2: $\neg A$ assumption
- 3: A A
- 4: \perp $\neg E$ 3,2
- 5: B $\perp-E$ 4
- 6: B assumption
- 7: B $\vee-E$ 1,2-5,6-6

Given:

$\neg A \vee B$
 A

FORMULAE 0 B , 1 $\neg A$, 2 A , 3 $\neg A \vee B$

SEQ

(cut«3,0/B,C»)
 (GIVEN 0)
 ("∨-E"«1,0,0/A,B,C»)
 (hyp«3/A»)
 ("⊥-E"«0/A»)
 ("¬-E"«2/B»)
 (GIVEN 1)
 (hyp«1/A»)
 (hyp«0/A»)

DERIVED RULE IS FROM $A \vee \neg B$ AND B INFER A

- 1: $A \vee \neg B$ $A \vee \neg B$
- 2: A assumption
- 3: $\neg B$ assumption
- 4: B B
- 5: \perp $\neg E$ 4,3
- 6: A $\perp E$ 5
- 7: A $\vee E$ 1,2-2,3-6

Given:
 $A \vee \neg B$
 B

FORMULAE 0 $\neg B$, 1 B , 2 A , 3 $A \vee \neg B$

SEQ
 (cut«3,2/B,C») (GIVEN 0)
 ("∨-E"«2,0,2/A,B,C») (hyp«3/A») (hyp«2/A») ("⊥-E"«2/A») ("¬-E"«1/B») (GIVEN 1) (hyp«0/A»)

DERIVED RULE IS FROM $A \wedge B$ INFER $A \wedge B$

- 1: $A \wedge B$ $A \wedge B$
- 2: if A then B else A fi $A \wedge B \equiv \text{if } A \text{ then } B \text{ else } A \text{ fi}$ 1
- 3: $A \vee \neg A$ FROM if A then B else C fi INFER $A \vee \neg A$ 2
- 4: A assumption
- 5: B FROM A INFER if A then B else C fi $\equiv B$ 4,2
- 6: $A \wedge B$ $\wedge I$ 4,5
- 7: $\neg A$ assumption
- 8: A FROM $\neg A$ INFER if A then B else C fi $\equiv C$ 7,2
- 9: \perp $\neg E$ 8,7
- 10: $A \wedge B$ $\perp E$ 9
- 11: $A \wedge B$ $\vee E$ 3,4-6,7-10

Given:
 $A \wedge B$

FORMULAE 0 $\neg A$, 1 A , 2 $A \wedge B$, 3 if A then B else A fi, 4 B , 5 $xx2$, 6 $xx1$, 7 $A \vee \neg A$, 8 $A \wedge B$, 9 xx

SEQ
 (cut«8,2/B,C») (GIVEN 0)

```

(cut«3,2/B,C»)
(LAYOUT "A^^B≐if A then B else A fi" ALL
  ("rewrite≐"«3,9,8,9/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«8,3/A,B»)
    (LAYOUT HIDEROOT
      ("A^^B≐if A then B else A fi"«1,4/A,B»)))
  (hyp«8/A»)
(cut«7,2/B,C»)
("FROM if A then B else C fi INFER A∨¬A"«1,4,1/A,B,C»)
(hyp«3/A»)
("∨-E"«1,0,2/A,B,C»)
(hyp«7/A»)
(cut«4,2/B,C»)
(LAYOUT "FROM A INFER if A then B else C fi≐B" ALL
  ("rewrite≐"«4,6,3,6/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«3,4/A,B»)
    (LAYOUT HIDEROOT
      ("FROM A INFER if A then B else C fi≐B"«1,4,1/A,B,C»)
      (hyp«1/A»)))
  (hyp«3/A»)
(cut«1,2/B,C»)
(hyp«1/A»)
(cut«4,2/B,C»)
(hyp«4/A»)
(LAYOUT COMPRESS "∧-I" ALL
  ("∧-I"«1,4/A,B»)
  (hyp«1/A»)
  (hyp«4/A»))
(cut«1,2/B,C»)
(LAYOUT "FROM ¬A INFER if A then B else C fi≐C" ALL
  ("rewrite≐"«1,5,3,5/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«3,1/A,B»)
    (LAYOUT HIDEROOT
      ("FROM ¬A INFER if A then B else C fi≐C"«1,4,1/A,B,C»)
      (hyp«0/A»)))
  (hyp«3/A»)
("⊥-E"«2/A»)
("¬-E"«1/B»)
(hyp«1/A»)
(hyp«0/A»)

```

 DERIVED RULE IS FROM $\neg(A \wedge B)$ INFER $\neg(A \wedge B)$

1:	$\neg(A \wedge B)$	$\neg(A \wedge B)$
2:	$A \wedge B$	assumption
3:	A	\wedge -E 2
4:	B	\wedge -E 2
5:	if A then B else A fi	FROM A INFER if A then B else C fi≐B 3,4
6:	$A \wedge B$	$A \wedge B \equiv$ if A then B else A fi 5
7:	\perp	\neg -E 6,1
8:	$\neg(A \wedge B)$	\neg I 2-7

Given:
 $\neg(A \wedge B)$

FORMULAE 0 $\neg(A \wedge B)$, 1 B, 2 A, 3 if A then B else A fi, 4 xx1, 5 $A \wedge B$, 6 xx, 7 $A \wedge B$, 8 \perp , 9 $\neg(A \wedge B)$

```

SEQ
(cut«0,9/B,C»)
(GIVEN 0)
("¬-I"«7/A»)
(cut«2,8/B,C»)
(LAYOUT "∧-E" ALL
  ("∧-E(L)"«1,2/B,A»)
  (hyp«7/A»)
(cut«1,8/B,C»)
(LAYOUT "∧-E" ALL
  ("∧-E(R)"«2,1/A,B»)
  (hyp«7/A»)
("¬-E"«5/B»)
(LAYOUT "A∧B≐if A then B else A fi" ALL
  ("rewrite≐"«5,6,3,6/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A∧B≐if A then B else A fi"«2,1/A,B»))
  (LAYOUT "FROM A INFER if A then B else C fi≐B" ALL
    ("rewrite≐"«3,4,1,4/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("FROM A INFER if A then B else C fi≐B"«2,1,2/A,B,C»)
      (hyp«2/A»)
      (hyp«1/A»)))
(hyp«0/A»)

```

DERIVED RULE IS FROM $A \vee B$ INFER $A \vee (\neg A \wedge B)$

1: $A \vee B$	$A \vee B$
2: if A then A else B fi	$A \vee B \equiv \text{if A then A else B fi}$ 1
3: $A \vee \neg A$	FROM if A then B else C fi INFER $A \vee \neg A$ 2
4: A	assumption
5: $A \vee (\neg A \wedge B)$	$\vee\text{-I(L)}$ 4
6: $\neg A$	assumption
7: B	FROM $\neg A$ INFER if A then B else C fi≐C 6,2
8: $\neg A \wedge B$	$\wedge\text{-I}$ 6,7
9: $A \vee (\neg A \wedge B)$	$\vee\text{-I(R)}$ 8
10: $A \vee (\neg A \wedge B)$	$\vee\text{-E}$ 3,4-5,6-9

Given:
 $A \vee B$

FORMULAE 0 B, 1 $\neg A$, 2 A, 3 $\neg A \wedge B$, 4 $A \vee (\neg A \wedge B)$, 5 if A then A else B fi, 6 xx1, 7 $A \vee \neg A$, 8 $A \vee B$, 9 xx

```

SEQ
(cut«8,4/B,C»)
(GIVEN 0)
(cut«5,4/B,C»)
(LAYOUT "A∨B≐if A then A else B fi" ALL
  ("rewrite≐"«5,9,8,9/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«8,5/A,B»)
    (LAYOUT HIDEROOT
      ("A∨B≐if A then A else B fi"«2,0/A,B»)))
  (hyp«8/A»)
(cut«7,4/B,C»)
("FROM if A then B else C fi INFER A∨¬A"«2,2,0/A,B,C»)
(hyp«5/A»)
("∨-E"«2,1,4/A,B,C»)

```

```

(hyp«7/A»)
("∨-I(L)"«3,2/B,A»)
(hyp«2/A»)
(cut«1,4/B,C»)
(hyp«1/A»)
(cut«0,4/B,C»)
(LAYOUT "FROM ¬A INFER if A then B else C fi≠C" ALL
  ("rewrite≠"«0,6,5,6/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≠"«5,0/A,B»)
    (LAYOUT HIDEROOT
      ("FROM ¬A INFER if A then B else C fi≠C"«2,2,0/A,B,C»)
      (hyp«1/A»)))
    (hyp«5/A»)
  (cut«0,4/B,C»)
  (hyp«0/A»)
  ("∨-I(R)"«2,3/A,B»)
  (LAYOUT COMPRESS "∧-I" ALL
    ("∧-I"«1,0/A,B»)
    (hyp«1/A»)
    (hyp«0/A»))

```

DERIVED RULE IS FROM $\neg(A \vee \vee B)$ INFER $\neg A \wedge \neg B$

1:	$\neg(A \vee \vee B)$	$\neg(A \vee \vee B)$
2:	$\neg(\text{if } A \text{ then } A \text{ else } B \text{ fi})$	$A \vee \vee B \triangleq \text{if } A \text{ then } A \text{ else } B \text{ fi } 1$
3:	A	assumption
4:	if A then A else B fi	FROM A INFER if A then B else C fi≠B 3,3
5:	⊥	¬-E 4,2
6:	$\neg A$	¬-I 3-5
7:	B	assumption
8:	if A then A else B fi	FROM ¬A INFER if A then B else C fi≠C 6,7
9:	⊥	¬-E 8,2
10:	$\neg B$	¬-I 7-9
11:	$\neg A \wedge \neg B$	\wedge -I 6,10

Given:
 $\neg(A \vee \vee B)$

FORMULAE 0 $\neg B$, 1 $\neg A$, 2 $\neg \text{if } A \text{ then } A \text{ else } B \text{ fi}$, 3 B, 4 A, 5 $\text{if } A \text{ then } A \text{ else } B \text{ fi}$, 6 $xx2$, 7 $\neg A \wedge \neg B$, 8 $xx1$,
 9 $\neg(A \vee \vee B)$, 10 $A \vee \vee B$, 11 xx , 12 $\neg(xx)$, 13 $\neg(\text{if } A \text{ then } A \text{ else } B \text{ fi})$

SEQ

```

(cut«9,7/B,C»)
(GIVEN 0)
(cut«13,7/B,C»)
(LAYOUT "A∨∨B≠if A then A else B fi" ALL
  ("rewrite≠"«5,11,10,12/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≠"«10,5/A,B»)
    (LAYOUT HIDEROOT
      ("A∨∨B≠if A then A else B fi"«4,3/A,B»)))
    (hyp«9/A»)
  (cut«1,7/B,C»)
  ("¬-I"«4/A»)
  ("¬-E"«5/B»)
  (LAYOUT "FROM A INFER if A then B else C fi≠B" ALL

```

```

("rewrite≐"«5,8,4,8/A,xx,B,P»)
(LAYOUT HIDEROOT
  ("FROM A INFER if A then B else C fi≐B"«4,4,3/A,B,C»)
  (hyp«4/A»)
  (hyp«4/A»)
  (hyp«2/A»)
  (cut«0,7/B,C»)
  ("¬-I"«3/A»)
  ("¬-E"«5/B»)
(LAYOUT "FROM ¬A INFER if A then B else C fi≐C" ALL
  ("rewrite≐"«5,6,3,6/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("FROM ¬A INFER if A then B else C fi≐C"«4,4,3/A,B,C»)
    (hyp«1/A»)
    (hyp«3/A»)
  (hyp«2/A»)
(LAYOUT COMPRESS "¬-I" ALL
  ("¬-I"«1,0/A,B»)
  (hyp«1/A»)
  (hyp«0/A»)

```

DERIVED RULE IS FROM $\neg(A \vee B)$ INFER $\neg A \wedge \neg B$

```

1:  $\neg(A \vee B)$   $\neg(A \vee B)$ 
2:  $A$  assumption
3:  $A \vee B$   $\vee$ -I(L) 2
4:  $\perp$   $\neg$ -E 3,1
5:  $\neg A$   $\neg$ -I 2-4
6:  $B$  assumption
7:  $A \vee B$   $\vee$ -I(R) 6
8:  $\perp$   $\neg$ -E 7,1
9:  $\neg B$   $\neg$ -I 6-8
10:  $\neg A \wedge \neg B$   $\wedge$ -I 5,9

```

Given:
 $\neg(A \vee B)$

FORMULAE 0 $\neg B$, 1 $\neg A$, 2 $\neg(A \vee B)$, 3 B , 4 A , 5 $A \vee B$, 6 $\neg A \wedge \neg B$

SEQ

```

(cut«2,6/B,C»)
(GIVEN 0)
(cut«1,6/B,C»)
("¬-I"«4/A»)
("¬-E"«5/B»)
("¬-I(L)"«3,4/B,A»)
(hyp«4/A»)
(hyp«2/A»)
(cut«0,6/B,C»)
("¬-I"«3/A»)
("¬-E"«5/B»)
("¬-I(R)"«4,3/A,B»)
(hyp«3/A»)
(hyp«2/A»)
(LAYOUT COMPRESS "¬-I" ALL
  ("¬-I"«1,0/A,B»)
  (hyp«1/A»)
  (hyp«0/A»)

```

DERIVED RULE IS FROM $\neg(A \wedge B)$ AND A INFER $\neg B$

- | | |
|-----------------------|--------------------|
| 1: B | assumption |
| 2: A | A |
| 3: $(A \wedge B)$ | \wedge -I 2,1 |
| 4: $\neg(A \wedge B)$ | $\neg(A \wedge B)$ |
| 5: \perp | \neg -E 3,4 |
| 6: $\neg B$ | \neg -I 1-5 |

Given:
 $\neg(A \wedge B)$
 A

FORMULAE 0 B, 1 A, 2 $(A \wedge B)$, 3 \perp

SEQ
 ("¬-I"«0/A») (cut«1,3/B,C») (GIVEN 1) (cut«0,3/B,C») (hyp«0/A») ("¬-E"«2/B») (LAYOUT COMPRESS "∧-I" ALL ("∧-I"«1,0/A,B») (hyp«1/A») (hyp«0/A») (GIVEN 0)

DERIVED RULE IS FROM $\neg(A \wedge B)$ AND B INFER $\neg A$

- | | |
|-----------------------|--------------------|
| 1: A | assumption |
| 2: B | B |
| 3: $(A \wedge B)$ | \wedge -I 1,2 |
| 4: $\neg(A \wedge B)$ | $\neg(A \wedge B)$ |
| 5: \perp | \neg -E 3,4 |
| 6: $\neg A$ | \neg -I 1-5 |

Given:
 $\neg(A \wedge B)$
 B

FORMULAE 0 B, 1 A, 2 $(A \wedge B)$, 3 \perp

SEQ
 ("¬-I"«1/A») (cut«1,3/B,C») (hyp«1/A») (cut«0,3/B,C») (GIVEN 1) ("¬-E"«2/B») (LAYOUT COMPRESS "∧-I" ALL ("∧-I"«1,0/A,B») (hyp«1/A»))

(hyp«0/A»)
 (GIVEN 0)

DERIVED RULE IS FROM $\neg A$ AND $\neg B$ INFER $\neg(A \vee B)$

- | | | |
|----|------------------|--------------------|
| 1: | $A \vee B$ | assumption |
| 2: | A | assumption |
| 3: | $\neg A$ | $\neg A$ |
| 4: | \perp | $\neg E$ 2,3 |
| 5: | B | assumption |
| 6: | $\neg B$ | $\neg B$ |
| 7: | \perp | $\neg E$ 5,6 |
| 8: | \perp | $\vee E$ 1,2-4,5-7 |
| 9: | $\neg(A \vee B)$ | $\neg I$ 1-8 |

Given:
 $\neg A$
 $\neg B$

FORMULAE 0 B, 1 A, 2 $A \vee B$, 3 \perp

SEQ
 ("¬I"«2/A»)
 ("∨-E"«1,0,3/A,B,C»)
 (hyp«2/A»)
 ("¬E"«1/B»)
 (hyp«1/A»)
 (GIVEN 0)
 ("¬E"«0/B»)
 (hyp«0/A»)
 (GIVEN 1)

DERIVED RULE IS FROM $A \rightarrow B$ AND $\neg B$ INFER $\neg A$

- | | | |
|----|-------------------|---------------------|
| 1: | A | assumption |
| 2: | $A \rightarrow B$ | $A \rightarrow B$ |
| 3: | B | $\rightarrow E$ 1,2 |
| 4: | $\neg B$ | $\neg B$ |
| 5: | \perp | $\neg E$ 3,4 |
| 6: | $\neg A$ | $\neg I$ 1-5 |

Given:
 $A \rightarrow B$
 $\neg B$

FORMULAE 0 A, 1 B

SEQ
 ("¬I"«0/A»)
 ("¬E"«1/B»)
 ("→E"«0,1/A,B»)
 (hyp«0/A»)

(GIVEN 0)
(GIVEN 1)

1.3. Collections: sequences, lists, maps, sets, trees

‘Pure’ – i.e. independent of pointer logic – treatment of sequences, lists and maps.

1.3.1. Definitions

Definitions are just rules.

1.3.1.1. Sequences

The empty sequence, append and zip behave as you would expect with any sequence.

RULE IS $S@() \triangleq S$
 RULE IS $()@S \triangleq S$
 RULE IS $R@S@T \triangleq R@(S@T)$
 RULE IS $S|||() \triangleq ()$
 RULE IS $()|||S \triangleq ()$
 RULE IS $((A)@R)|||((B)@S) \triangleq ((A,B))@(R|||S)$

Reverse works for finite sequences.

RULE IS $\text{rev } () \triangleq ()$
 RULE IS $\text{rev } (A) \triangleq (A)$
 RULE IS FROM finitesequence R AND finitesequence S INFER $\text{rev}(R@S) \triangleq \text{rev } S @ \text{rev } R$

Maybe some of the definitions of perm need finitesequence qualifications.

RULE IS $\text{perm}(S,S)$
 RULE IS $\text{perm}(S1@(A)@S2@S3, S1@S2@(A)@S3)$
 RULE IS $\text{perm}(R,S) \triangleq \text{perm}(S,R)$
 RULE IS FROM $\text{perm}(R,S)$ AND $\text{perm}(S,T)$ INFER $\text{perm}(R,T)$

Arithmetic is difficult to formalise (never mind completeness, it’s hard work to get anywhere). These rules are included to make some of the invariant proofs go through.

RULE IS $\text{length}(R@(A)@S) > 0$
 RULE IS FROM finitesequence R AND finitesequence S INFER $\text{length}(R@(A)@S) > \text{length}(R@S)$

More about append.

RULE IS $R@(A)@T \neq ()$
 RULE IS FROM finitesequence R AND finitesequence T INFER $R@(A)@T \neq R@T$
 RULE IS $(A)@R=(B)@S \triangleq A=B \wedge R=S$
 RULE IS $R@(A)=S@(B) \triangleq R=S \wedge A=B$

1.3.1.2. . Ordered sequences

Not entirely a pure definition, because it depends on cells in the heap, pointed to by the elements of the sequence.

RULE IS $\text{oseq } (E,())$
 RULE IS $\text{oseq } (E,(A))$
 RULE IS $\text{oseq } (E,R@(A)@(B)@S) \triangleq \text{oseq}(E,R@(A)) \wedge A.E \leq B.E \wedge \text{oseq}(E,(B)@S)$

1.3.1.3. Sequence comprehensions

RULE IS $\{ A \mid x \in (B)@S \} \triangleq (A)\langle B/x \rangle @ \{ A \mid x \in S \}$
 RULE IS $\{ A \mid x \in () \} \triangleq ()$
 RULE IS $\{ A \mid x \in (B)@S \mid C \} \triangleq \text{if } C \langle B/x \rangle \text{ then } (A)\langle B/x \rangle \text{ else } () \text{ fi} @ \{ A \mid x \in S \mid C \}$
 RULE IS $\{ A \mid x \in () \mid C \} \triangleq ()$

1.3.1.4. Separation

Separation turns out to be crucial to the proofs. The non-intersection operator $\neg \cap$ is my treatment of it.

RULE IS $S \neg \neg ()$
 RULE IS $() \neg \neg S$
 RULE IS $R \neg \neg S \triangleq S \neg \neg R$
 RULE IS $R \neg \neg S @ T \triangleq R \neg \neg S \wedge R \neg \neg T$
 RULE IS $R @ S \neg \neg T \triangleq R \neg \neg T \wedge S \neg \neg T$
 RULE IS $(A) \neg \neg (B) \triangleq A \neq B$

1.3.1.5. Finite sequences

Finite sequences are special.

RULE IS finitesequence $()$
 RULE IS finitesequence (A)
 RULE IS finitesequence $(R @ S) \triangleq$ finitesequence $R \wedge$ finitesequence S

We allow induction over finite sequences.

RULE "finite sequence induction (L)"(OBJECT y , OBJECT ys) WHERE FRESH y , ys IS
 FROM $P \ll () / xs \gg$ AND $P \ll ys / xs \gg \vdash P \ll (y) @ ys / xs \gg$ INFER $\forall xs : (\text{finitesequence } xs \rightarrow P)$
 RULE "finite sequence induction (R)"(OBJECT y , OBJECT ys) WHERE FRESH y , ys IS
 FROM $P \ll () / xs \gg$ AND $P \ll ys / xs \gg \vdash P \ll ys @ (y) / xs \gg$ INFER $\forall xs : (\text{finitesequence } xs \rightarrow P)$

1.3.1.6. Lists

Lists are finite sequences without repetitions.

RULE IS list $()$
 RULE IS list (A)
 RULE IS list $(R @ S) \triangleq$ list $R \wedge$ list $S \wedge R \neg \neg S$

This is a definition, but could it be proved by induction?

RULE IS FROM list S INFER finitesequence S

1.3.1.7. Ordered lists

RULE IS olist $(E, S) \triangleq$ oseq $(E, S) \wedge$ list S

1.3.1.8. Mappings

RULE IS $A.(B @ C \rightarrow E) \triangleq$ if $A=C$ then E else $A.B$ fi

1.3.1.9. Sets

We can talk about the set represented by a sequence. Non-intersection of sequences has an obvious relationship with non-membership of a sequence-set.

RULE IS set $() \triangleq \{\}$
 RULE IS set $(A) \triangleq \{A\}$
 RULE IS set $(R @ S) \triangleq$ set $R \cup$ set S
 RULE IS $(A) \neg \neg S \triangleq \neg(A \in \text{set } S)$

Set operators

RULE IS $S \cup \{\}$ $\triangleq S$
 RULE IS $\{\} \cup S \triangleq S$
 RULE IS $R \cup S \triangleq S \cup R$
 RULE IS $S \cup S \triangleq S$
 RULE " $(R \cup S) \cup T \triangleq R \cup (S \cup T)$ " IS $R \cup S \cup T \triangleq R \cup (S \cup T)$
 RULE (OBJECT x) IS $A \subseteq B \triangleq \forall x: (x \in A \rightarrow x \in B)$
 RULE IS $S = T \triangleq S \subseteq T \wedge T \subseteq S$
 RULE IS $\neg(A \in \{\})$
 RULE IS $A \in S \vee \neg(A \in S)$ /* \in is a kind of equality operator, so I think this is ok */
 RULE IS $A \in \{B\} \triangleq A = B$
 RULE IS $A \in R \cup S \triangleq A \in R \vee A \in S$

\cup is big union.

```

RULE IS U {} ≐ {}
RULE IS U({A}∪B) ≐ A∪B
RULE IS SU{A} ≠ {}

```

Finite sets are useful not least because they allow induction. (I think we only deal with finite sets, but then again you never know.)

```

RULE IS finiteset {}
RULE IS finiteset {A}
RULE IS finiteset(R∪S) ≐ finiteset R ∧ finiteset S

RULE "finite set induction" (OBJECT x, OBJECT ys, OBJECT zs) WHERE FRESH x, ys, zs IS
  FROM P«{/xs» AND P«{x}/xs» AND P«ys/xs»,P«zs/xs» ⊢ P«ys∪zs/xs»
  INFER ∀xs:(finiteset xs → P)

```

1.3.1.10. Set comprehensions

```

RULE IS { A | x∈{B}∪S } ≐ {A}«B/x»∪{ A | x∈S }
RULE IS { A | x∈{} } ≐ {}
RULE IS { A | x∈{B}∪S | C } ≐ if C«B/x» then {A}«B/x» else {} fi∪{ A | x∈S | C }
RULE IS { A | x∈{} | C } ≐ {}

```

Equality in set comprehensions is a rule. Equality in sequence comprehensions is a derived rule (see below).

```

RULE IS FROM ∀x:(x∈S → A=A') INFER { A | x∈S } = { A' | x∈S }

```

1.3.1.11. Tuples

Equality in tuples. Once again, you have to list all the variant forms you are going to use

```

RULE IS (A1,B1)=(A2,B2) ≐ A1=A2 ∧ B1=B2
RULE IS (A1,B1,C1)=(A2,B2,C2) ≐ A1=A2 ∧ B1=B2 ∧ C1=C2

```

1.3.1.12. Trees

Triples represent binary trees: A is the root, B and C the subtrees. The empty tree is the empty tuple. For some reason I've defined equality again ...

```

RULE IS (A,TA1,TA2)=(B,TB1,TB2) ≐ A=B ∧ TA1=TB1 ∧ TA2=TB2
RULE IS ¬((A,T1,T2)=()) /* ready for definedness */

```

Trees of finite height admit induction.

```

RULE IS finiteheight ()
RULE IS finiteheight (A,B,C) ≐ finiteheight B ∧ finiteheight C

RULE "binary tree induction"(OBJECT yt, OBJECT zt, OBJECT x) WHERE FRESH x,yt,zt IS
  FROM P«()/xt» AND P«yt/xt», P«zt/xt» ⊢ P«(x,yt,zt)/xt»
  INFER ∀xt:(finiteheight xt → P)

```

We can talk about the set of nodes in a tree.

```

RULE IS set() ≐ {}
RULE IS set(A,B,C) ≐ {A}∪set B∪set C

```

1.3.2. Menus, panels etc.

The Sequences panel has some mechanism which can flatten (eliminate brackets from) append formulæ, and a couple of axioms which are always hidden in proofs.

```

ENTRY "flatten @" IS
  iterateR2L "rewrite≐" "symmetric≐" (QUOTE (_R@(_S@_T))) "(R@S)@T≐R@(S@T)"
  (Fail "no @s to flatten")

ENTRY "list ()" IS LAYOUT HIDEROOT "list()"
ENTRY "list (A)" IS LAYOUT HIDEROOT "list(A)"

```

The Instructions panel contains mechanism which allows a semicolon sequence to be taken apart in a single step, strengthening postcondition steps to be hidden and semicolon formulæ to be flattened.

```
ENTRY "S1;S2" IS
  (LETGOALPATH G (LAYOUT COMPRESS "sequence")
    semicolon
    (GOALPATH (SUBGOAL G 1)))
ENTRY "strengthen postcondition" IS
  (LETGOALPATH G "strengthen postcondition"
    (GOALPATH (SUBGOAL G 1))
    (LAYOUT HIDEROOT "→-!"))
ENTRY "flatten ;" IS
  iterateR2L "rewrite≐" "symmetric≐" (QUOTE (_A;(_B;_C)))
    "(A;B);C≐A;(B;C)" (Fail "no semicolons to flatten")
```

There's an entry in the Sets panel to flatten unions.

```
ENTRY "flatten ∪" IS
  iterateR2L "rewrite≐" "symmetric≐" (QUOTE (_R∪(_S∪_T))) "(R∪S)∪T≐R∪(S∪T)"
    (Fail "no ∪s to flatten")
```

1.3.3. Proofs

1.3.3.1. Maps

DERIVED RULE IS $A.(B \oplus A \mapsto E) = E$

1: if $A=A$ then E else $A.B$ fi= E FROM A INFER if A then B else C fi= B

2: $A.(B \oplus A \mapsto E)=E$ $A.(B \oplus C \mapsto E) \equiv$ if $A=C$ then E else $A.B$ fi 1

FORMULAE 0 E , 1 A , 2 $A=A$, 3 $A.B$, 4 if $A=A$ then E else $A.B$ fi, 5 $xx1$, 6 $xx1=E$, 7 B , 8 $A.(B \oplus A \mapsto E)$, 9 xx , 10 $xx=E$

```
LAYOUT "A.(B⊕C→E)≐if A=C then E else A.B fi" ALL
  ("rewrite≐"«8,9,4,10/A,xx,B,P»)
(LAYOUT HIDEROOT
  ("A.(B⊕C→E)≐if A=C then E else A.B fi"«1,7,1,0/A,B,C,E»))
(LAYOUT "FROM A INFER if A then B else C fi≐B" ALL
  ("rewrite≐"«4,5,0,6/A,xx,B,P»)
(LAYOUT HIDEROOT
  ("FROM A INFER if A then B else C fi≐B"«2,0,3/A,B,C»)
(LAYOUT HIDEROOT
  ("reflexive="«1/A»)))
(LAYOUT HIDEROOT
  ("reflexive="«0/A»)))
```

DERIVED RULE IS FROM $A=C$ INFER $A.(B \oplus C \mapsto E) = E$

1: $A=C$ $A=C$

2: if $A=C$ then E else $A.B$ fi= E FROM A INFER if A then B else C fi= B 1

3: $A.(B \oplus C \mapsto E)=E$ $A.(B \oplus C \mapsto E) \equiv$ if $A=C$ then E else $A.B$ fi 2

Given:

$A=C$

FORMULAE 0 E , 1 $A=C$, 2 $A.B$, 3 if $A=C$ then E else $A.B$ fi, 4 $xx2$, 5 $xx2=E$, 6 A , 7 B , 8 C , 9 $A.(B \oplus C \mapsto E)$, 10 xx , 11 $xx=E$

```
LAYOUT "A.(B⊕C→E)≐if A=C then E else A.B fi" ALL
  ("rewrite≐"«9,10,3,11/A,xx,B,P»)
(LAYOUT HIDEROOT
  ("A.(B⊕C→E)≐if A=C then E else A.B fi"«6,7,8,0/A,B,C,E»))
(LAYOUT "FROM A INFER if A then B else C fi≐B" ALL
```

```

("rewrite≐"«3,4,0,5/A,xx,B,P»)
(LAYOUT HIDEROOT
  ("FROM A INFER if A then B else C fi≐B"«1,0,2/A,B,C»)
  (GIVEN 0))
(LAYOUT HIDEROOT
  ("reflexive="«0/A»)))

```

DERIVED RULE IS FROM $A \neq C$ INFER $A.(B \oplus C \rightarrow E) = A.B$

1: $A \neq C$	$A \neq C$
2: $\neg(A=C)$	$\neg(A=B) \neq A \neq B$ 1
3: if $A=C$ then E else $A.B$ fi= $A.B$	FROM $\neg A$ INFER if A then B else C fi= C 2
4: $A.(B \oplus C \rightarrow E) = A.B$	$A.(B \oplus C \rightarrow E) \neq$ if $A=C$ then E else $A.B$ fi 3

Given:
 $A \neq C$

FORMULAE 0 $A.B$, 1 A , 2 C , 3 $\neg(A=C)$, 4 $xx2$, 5 $A \neq C$, 6 $A=C$, 7 E , 8 if $A=C$ then E else $A.B$ fi, 9 $xx1$, 10 $xx1=A.B$, 11 B , 12 $A.(B \oplus C \rightarrow E)$, 13 xx , 14 $xx=A.B$

```

LAYOUT "A.(B⊕C→E)≐if A=C then E else A.B fi" ALL
("rewrite≐"«12,13,8,14/A,xx,B,P»)
(LAYOUT HIDEROOT
  ("A.(B⊕C→E)≐if A=C then E else A.B fi"«1,11,2,7/A,B,C,E»))
(LAYOUT "FROM ¬A INFER if A then B else C fi≐C" ALL
  ("rewrite≐"«8,9,0,10/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("FROM ¬A INFER if A then B else C fi≐C"«6,7,0/A,B,C»)
    (LAYOUT "¬(A=B)≐A≠B" ALL
      ("rewrite≐"«3,4,5,4/A,xx,B,P»)
      (LAYOUT HIDEROOT
        ("¬(A=B)≐A≠B"«1,2/A,B»)
        (GIVEN 0)))
    (LAYOUT HIDEROOT
      ("reflexive="«0/A»)))

```

1.3.3.2. Lists

DERIVED RULE IS FROM $\text{list}((A)@S)$ INFER $(A)\neg\cap S$

1: $\text{list}((A)@S)$	$\text{list}((A)@S)$
2: $\text{list}(A) \wedge \text{list } S \wedge (A)\neg\cap S$	$\text{list}(R@S) \neq \text{list } R \wedge \text{list } S \wedge R\neg\cap S$ 1
3: $\text{list}(A)$	\wedge -E 2
4: $\text{list } S$	\wedge -E 2
5: $(A)\neg\cap S$	\wedge -E 2

Given:
 $\text{list}((A)@S)$

FORMULAE 0 $(A)\neg\cap S$, 1 $\text{list}(A) \wedge \text{list } S \wedge (A)\neg\cap S$, 2 $\text{list}(A) \wedge \text{list } S$, 3 $\text{list}(A)$, 4 $\text{list } S$, 5 $\text{list}((A)@S)$, 6 (A) , 7 S , 8 xx

```

SEQ
(cut«5,0/B,C»)
(GIVEN 0)
(cut«1,0/B,C»)
(LAYOUT "list(R@S)≐list R∧list S∧R¬∩S" ALL
  ("rewrite≐"«1,8,5,8/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«5,1/A,B»)
    (LAYOUT HIDEROOT

```

```

("list(R@S)≠list R∧list S∧R¬⊥S"«6,7/R,S»))
(hyp«5/A»)
(LAYOUT HIDE CUT
(cut«2,0/B,C»)
("∧-E(L)"«0,2/B,A»)
(hyp«1/A»)
(cut«3,0/B,C»)
(LAYOUT "∧-E" ALL
("∧-E(L)"«4,3/B,A»)
(hyp«2/A»)
(cut«4,0/B,C»)
(LAYOUT "∧-E" ALL
("∧-E(R)"«3,4/A,B»)
(hyp«2/A»)
(cut«0,0/B,C»)
(LAYOUT "∧-E" ALL
("∧-E(R)"«2,0/A,B»)
(hyp«1/A»)
(hyp«0/A»)

```

DERIVED RULE IS FROM list S INFER list(⊥@S)

- 1: list S list S
- 2: list(⊥@S) ⊥@S ≐ S 1

Given:
list S

FORMULAE 0 list(⊥@S), 1 list S, 2 S, 3 ⊥@S, 4 xx, 5 list xx

SEQ

```

(cut«1,0/B,C»)
(GIVEN 0)
(cut«0,0/B,C»)
(LAYOUT "⊥@S≐S" ALL
("rewrite≐"«3,4,2,5/A,xx,B,P»)
(LAYOUT HIDER ROOT
("⊥@S≐S"«2/S»)
(hyp«1/A»)
(hyp«0/A»)

```

DERIVED RULE IS FROM list S INFER list(S@⊥)

- 1: list(S) list S
- 2: list(S@⊥) S@⊥ ≐ S 1

Given:
list S

FORMULAE 0 S, 1 S@⊥, 2 xx, 3 list(xx)

```

LAYOUT "S@⊥≐S" ALL
("rewrite≐"«1,2,0,3/A,xx,B,P»)
(LAYOUT HIDER ROOT
("S@⊥≐S"«0/S»)
(GIVEN 0)

```

1.3.3.3. Sequences

DERIVED RULE IS $\langle A \rangle \neq \langle \rangle$

1: $\langle \rangle @ \langle A \rangle @ \langle \rangle \neq \langle \rangle$ $R @ \langle A \rangle @ T \neq \langle \rangle$

2: $\langle \rangle @ \langle A \rangle \neq \langle \rangle$ $S @ \langle \rangle \neq S$ 1

3: $\langle A \rangle \neq \langle \rangle$ $\langle \rangle @ S \neq S$ 2

FORMULAE 0 A, 1 $\langle \rangle$, 2 $\langle \rangle @ \langle A \rangle$, 3 $\langle \rangle @ \langle A \rangle @ \langle \rangle$, 4 xx, 5 $xx \neq \langle \rangle$, 6 $\langle A \rangle$, 7 $xx3$, 8 $xx3 \neq \langle \rangle$

LAYOUT " $\langle \rangle @ S \neq S$ " ALL

("rewrite" «6,7,2,8/A,xx,B,P»)

(LAYOUT HIDEROOT

("symmetric" «2,6/A,B»)

(LAYOUT HIDEROOT

("" $\langle \rangle @ S \neq S$ " «6/S»)))

(LAYOUT " $S @ \langle \rangle \neq S$ " ALL

("rewrite" «2,4,3,5/A,xx,B,P»)

(LAYOUT HIDEROOT

("symmetric" «3,2/A,B»)

(LAYOUT HIDEROOT

("" $S @ \langle \rangle \neq S$ " «2/S»)))

("" $R @ \langle A \rangle @ T \neq \langle \rangle$ " «0,1,1/A,R,T»))

DERIVED RULE IS $\langle A \rangle @ T \neq \langle \rangle$

1: $\langle \rangle @ \langle A \rangle @ T \neq \langle \rangle$ $R @ \langle A \rangle @ T \neq \langle \rangle$

2: $\langle A \rangle @ T \neq \langle \rangle$ $\langle \rangle @ S \neq S$ 1

FORMULAE 0 A, 1 $\langle \rangle$, 2 T, 3 $\langle A \rangle$, 4 $\langle \rangle @ \langle A \rangle$, 5 xx, 6 $xx @ T \neq \langle \rangle$

LAYOUT " $\langle \rangle @ S \neq S$ " ALL

("rewrite" «3,5,4,6/A,xx,B,P»)

(LAYOUT HIDEROOT

("symmetric" «4,3/A,B»)

(LAYOUT HIDEROOT

("" $\langle \rangle @ S \neq S$ " «3/S»)))

("" $R @ \langle A \rangle @ T \neq \langle \rangle$ " «0,1,2/A,R,T»))

DERIVED RULE IS $R @ \langle A \rangle \neq \langle \rangle$

1: $R @ \langle A \rangle @ \langle \rangle \neq \langle \rangle$ $R @ \langle A \rangle @ T \neq \langle \rangle$

2: $R @ \langle A \rangle \neq \langle \rangle$ $S @ \langle \rangle \neq S$ 1

FORMULAE 0 A, 1 R, 2 $\langle \rangle$, 3 $R @ \langle A \rangle$, 4 $R @ \langle A \rangle @ \langle \rangle$, 5 xx, 6 $xx \neq \langle \rangle$

LAYOUT " $S @ \langle \rangle \neq S$ " ALL

("rewrite" «3,5,4,6/A,xx,B,P»)

(LAYOUT HIDEROOT

("symmetric" «4,3/A,B»)

(LAYOUT HIDEROOT

("" $S @ \langle \rangle \neq S$ " «3/S»)))

("" $R @ \langle A \rangle @ T \neq \langle \rangle$ " «0,1,2/A,R,T»))

DERIVED RULE IS FROM finitesequence R INFER $R@A \neq R$

1: finitesequence R	finitesequence R
2: $\langle \rangle @A \neq \langle \rangle$	Derived Rule $R@A \neq \langle \rangle$
3: $ys@A \neq ys$	assumption
4: $(y)@ys@A = (y)@ys$	assumption
5: $(y)@(ys@A) = (y)@ys$	$(R@S)@T \neq R@(S@T)$ 4
6: $y = y \wedge ys@A = ys$	$(A)@R = (B)@S \neq A = B \wedge R = S$ 5
7: $y = y$	\wedge -E 6
8: $ys@A = ys$	\wedge -E 6
9: $\neg(ys@A = ys)$	$\neg(A=B) \neq A \neq B$ 3
10: \perp	\neg -E 8,9
11: $\neg((y)@ys@A = (y)@ys)$	\neg -I 4-10
12: $(y)@ys@A \neq (y)@ys$	$\neg(A=B) \neq A \neq B$ 11
13: $\forall xs: (\text{finitesequence } xs \rightarrow xs@A \neq xs)$	finite sequence induction (L) 2,3-12
14: finitesequence $R \rightarrow R@A \neq R$	\forall -E 13
15: $R@A \neq R$	\rightarrow -E 1,14

Given:

finitesequence R

FORMULAE 0 $ys@A \neq ys$, 1 $ys@A$, 2 ys , 3 $\neg(ys@A = ys)$, 4 $xx3$, 5 $ys@A = ys$, 6 $y = y \wedge ys@A = ys$, 7 $y = y$, 8 \perp , 9 $(y)@(ys@A) = (y)@ys$, 10 y , 11 $xx2$, 12 $(y)@ys@A = (y)@ys$, 13 (y) , 14 (A) , 15 $(y)@ys@A$, 16 $(y)@(ys@A)$, 17 $xx1$, 18 $xx1 = (y)@ys$, 19 $(y)@ys$, 20 $(y)@ys@A \neq (y)@ys$, 21 $\neg((y)@ys@A = (y)@ys)$, 22 xx , 23 A , 24 $\langle \rangle$, 25 $xs@A \neq xs$, 26 xs , 27 R , 28 finitesequence $xs \rightarrow xs@A \neq xs$, 29 finitesequence R, 30 $R@A \neq R$

SEQ

```

("→-E"«29,30/A,B»)
(GIVEN 0)
("∀-E"«27,28,26/B,A,x»)
("finite sequence induction (L)"«10,2,25,26/y,ys,P,xs»)
("R@A≠⟨⟩"«23,24/A,R»)
(LAYOUT "¬(A=B)≠A≠B" ALL
  ("rewrite≠"«20,22,21,22/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≠"«20,21/B,A»)
    (LAYOUT HIDEROOT
      ("¬(A=B)≠A≠B"«15,19/A,B»)))
  ("¬-I"«12/A»)
  (cut«9,8/B,C»)
  (LAYOUT "(R@S)@T≠R@(S@T)" ALL
    ("rewrite="«16,17,15,18/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("symmetric≠"«15,16/A,B»)
      (LAYOUT HIDEROOT
        ("(R@S)@T≠R@(S@T)"«13,2,14/R,S,T»)))
    (hyp«12/A»)
    (cut«6,8/B,C»)
  (LAYOUT "(A)@R=(B)@S≠A=B∧R=S" ALL
    ("rewrite="«6,11,9,11/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("symmetric≠"«9,6/A,B»)
      (LAYOUT HIDEROOT

```

```

    ("(A)@R=(B)@S≐A=B∧R=S"«10,10,1,2/A,B,R,S»)))
    (hyp«9/A»)
    (cut«7,8/B,C»)
    (LAYOUT "∧-E" ALL
      ("∧-E(L)"«5,7/B,A»)
      (hyp«6/A»)
      (cut«5,8/B,C»)
      (LAYOUT "∧-E" ALL
        ("∧-E(R)"«7,5/A,B»)
        (hyp«6/A»)
        ("¬-E"«5/B»)
        (hyp«5/A»)
        (LAYOUT "¬(A=B)≐A≠B" ALL
          ("rewrite≐"«3,4,0,4/A,xx,B,P»)
          (LAYOUT HIDEROOT
            ("¬(A=B)≐A≠B"«1,2/A,B»)
            (hyp«0/A»)))
  
```

DERIVED RULE IS FROM finitesequence T INFER (A)@T ≠ T

1: finitesequence T	finitesequence T
2: (A)@()≠()	Derived Rule (A)@T≠()
3: (A)@ys≠ys	assumption
4: (A)@ys@(y)=ys@(y)	assumption
5: (A)@ys=ys∧y=y	R@(A)=S@(B)≐R=S∧A=B 4
6: (A)@ys=ys	∧-E(L) 5
7: ¬((A)@ys=ys)	¬(A=B)≐A≠B 3
8: ⊥	¬-E 6,7
9: ¬((A)@ys@(y)=ys@(y))	¬-I 4-8
10: (A)@ys@(y)≠ys@(y)	¬(A=B)≐A≠B 9
11: (A)@(ys@(y))≠ys@(y)	iterate 10
12: ∀xs:(finitesequence xs→(A)@xs≠xs)	finite sequence induction (R) 2,3-11
13: finitesequence T→(A)@T≠T	∀-E 12
14: (A)@T≠T	→-E 1,13

Given:
finitesequence T

FORMULAE 0 (A)@ys≠ys, 1 (A)@ys, 2 ys, 3 ¬((A)@ys=ys), 4 xx3, 5 (A)@ys=ys, 6 (A)@ys=ys∧y=y, 7 y=y, 8 ⊥, 9 (A)@ys@(y)=ys@(y), 10 y, 11 xx2, 12 (A)@ys@(y), 13 ys@(y), 14 (A)@ys@(y)≠ys@(y), 15 ¬((A)@ys@(y)=ys@(y)), 16 xx1, 17 (A), 18 (y), 19 (A)@(ys@(y)), 20 xx, 21 xx≠ys@(y), 22 A, 23 (), 24 (A)@xs≠xs, 25 xs, 26 T, 27 finitesequence xs→(A)@xs≠xs, 28 finitesequence T, 29 (A)@T≠T

SEQ

```

    ("→-E"«28,29/A,B»)
    (GIVEN 0)
    ("∀-E"«26,27,25/B,A,x»)
    ("finite sequence induction (R)"«10,2,24,25/y,ys,P,xs»)
    ("(A)@T≠()"«22,23/A,T»)
    (LAYOUT HIDEROOT
      (LAYOUT "iterate" (1)
        ("rewrite≐"«19,20,12,21/A,xx,B,P»)
        ("symmetric≐"«12,19/A,B»)
      )
    )
  
```

```

("R@S)@T≐R@(S@T)"«17,2,18/R,S,T»)
(LAYOUT "¬(A=B)≐A≠B" ALL
  ("rewrite≐"«14,16,15,16/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«14,15/B,A»)
    (LAYOUT HIDEROOT
      ("¬(A=B)≐A≠B"«12,13/A,B»)))
  ("¬-I"«9/A»)
  (cut«6,8/B,C»)
  (LAYOUT "R@{A}=S@{B}≐R=S^A=B" ALL
    ("rewrite≐"«6,11,9,11/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("symmetric≐"«9,6/A,B»)
      (LAYOUT HIDEROOT
        ("R@{A}=S@{B}≐R=S^A=B"«10,10,1,2/A,B,R,S»)))
    (hyp«9/A»)
  (cut«5,8/B,C»)
  ("^E(L)"«7,5/B,A»)
  (hyp«6/A»)
  ("¬-E"«5/B»)
  (hyp«5/A»)
  (LAYOUT "¬(A=B)≐A≠B" ALL
    ("rewrite≐"«3,4,0,4/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("¬(A=B)≐A≠B"«1,2/A,B»)
      (hyp«0/A»))))

```

1.3.3.4. Sequence comprehensions

DERIVED RULE IS FROM finitesequence S AND $\forall x:(x \in \text{set } S \rightarrow A=A')$ INFER $\langle A \mid x \in S \rangle = \langle A' \mid x \in S \rangle$

1: finitesequence S	finitesequence S
2: $\forall x:(x \in \text{set } S \rightarrow A=A')$	$\forall x:(x \in \text{set } S \rightarrow A=A')$
3: $\forall x:(x \in \text{set } () \rightarrow A=A')$	assumption
4: $\langle () \rangle = \langle A' \mid x \in () \rangle$	$\langle A \mid x \in () \rangle \triangleq ()$
5: $\langle A \mid x \in () \rangle = \langle A' \mid x \in () \rangle$	$\langle A \mid x \in () \rangle \triangleq ()$ 4
6: $\forall x:(x \in \text{set } () \rightarrow A=A') \rightarrow \langle A \mid x \in () \rangle = \langle A' \mid x \in () \rangle$	\rightarrow -I 3-5
7: $\forall x:(x \in \text{set } ys \rightarrow A=A') \rightarrow \langle A \mid x \in ys \rangle = \langle A' \mid x \in ys \rangle$	assumption
8: $\forall x:(x \in \text{set } ((y)@ys) \rightarrow A=A')$	assumption
9: $y \in \text{set } ((y)@ys) \rightarrow A \langle y/x \rangle = A' \langle y/x \rangle$	\forall -E 8
10: $y \in \{y\}$	$A \in \{B\} \triangleq A=B$
11: $y \in \{y\} \cup \text{set } ys$	Conjectured Rule FROM $A \in S$ INFER $A \in S \cup T$ 10
12: $y \in \text{set } (y) \cup \text{set } ys$	$\text{set } (A) \triangleq \{A\}$ 11
13: $y \in \text{set } ((y)@ys)$	$\text{set } (R@S) \triangleq \text{set } R \cup \text{set } S$ 12
14: $A \langle y/x \rangle = A' \langle y/x \rangle$	\rightarrow -E 13,9
15: $cc \in \text{set } ys$	assumption
16: $cc \in \text{set } ((y)@ys) \rightarrow A \langle cc/x \rangle = A' \langle cc/x \rangle$	\forall -E 8
17: $cc \in \text{set } (y) \cup \text{set } ys$	Conjectured Rule FROM $A \in T$ INFER $A \in S \cup T$ 15
18: $cc \in \text{set } ((y)@ys)$	$\text{set } (R@S) \triangleq \text{set } R \cup \text{set } S$ 17
19: $A \langle cc/x \rangle = A' \langle cc/x \rangle$	\rightarrow -E 18,16
20: $cc \in \text{set } ys \rightarrow A \langle cc/x \rangle = A' \langle cc/x \rangle$	\rightarrow -I 15-19
21: $\forall x:(x \in \text{set } ys \rightarrow A=A')$	\forall -I 20
22: $\langle A \mid x \in ys \rangle = \langle A' \mid x \in ys \rangle$	\rightarrow -E 21,7
23: $A \langle y/x \rangle = A' \langle y/x \rangle \wedge \langle A \mid x \in ys \rangle = \langle A' \mid x \in ys \rangle$	\wedge -I 14,22
24: $\langle A \langle y/x \rangle \rangle @ \langle A \mid x \in ys \rangle = \langle A' \langle y/x \rangle \rangle @ \langle A' \mid x \in ys \rangle$	$\langle A \rangle @ R = \langle B \rangle @ S \triangleq A=B \wedge R=S$ 23
25: $\langle A \langle y/x \rangle \rangle @ \langle A \mid x \in ys \rangle = \langle A' \mid x \in (y)@ys \rangle$	$\langle A \mid x \in (B)@S \rangle \triangleq \langle A \rangle \langle B/x \rangle @ \langle A \mid x \in S \rangle$ 24
26: $\langle A \mid x \in (y)@ys \rangle = \langle A' \mid x \in (y)@ys \rangle$	$\langle A \mid x \in (B)@S \rangle \triangleq \langle A \rangle \langle B/x \rangle @ \langle A \mid x \in S \rangle$ 25
27: $\forall x:(x \in \text{set } ((y)@ys) \rightarrow A=A') \rightarrow \langle A \mid x \in (y)@ys \rangle = \langle A' \mid x \in (y)@ys \rangle$	\rightarrow -I 8-26
28: $\forall xs:(\text{finitesequence } xs \rightarrow \forall x:(x \in \text{set } xs \rightarrow A=A')) \rightarrow \langle A \mid x \in xs \rangle = \langle A' \mid x \in xs \rangle$	finite sequence induction (L) 6,7-27
29: finitesequence S $\rightarrow \forall x:(x \in \text{set } S \rightarrow A=A') \rightarrow \langle A \mid x \in S \rangle = \langle A' \mid x \in S \rangle$	\forall -E 28
30: $\forall x:(x \in \text{set } S \rightarrow A=A') \rightarrow \langle A \mid x \in S \rangle = \langle A' \mid x \in S \rangle$	\rightarrow -E 1,29
31: $\langle A \mid x \in S \rangle = \langle A' \mid x \in S \rangle$	\rightarrow -E 2,30

Given:

finitesequence S

$\forall x:(x \in \text{set } S \rightarrow A=A')$

Provided:

x NOTIN S

The proof needs x NOTIN S but the theorem doesn't specify it. I didn't notice, and neither did Jape (that looks like a bug). Although formally there's no problem, in truth no harm's been done: I ought to have defined bounded quantification – by writing

BIND x SCOPE B in $\forall x \in A:B$

RULE " \forall -E"(C) IS FROM $\forall x \in A:B$ AND $C \in A$ INFER $B \langle C/x \rangle$

RULE " \forall -I"(OBJECT cc) WHERE FRESH cc IS FROM $cc \in A \wedge B \langle cc/x \rangle$ INFER $\forall x \in A:B$

– then S wouldn't have been in the scope of $\forall x$, and the theorem would have applied without the proviso.

FORMULAE 0 $(\text{Alx}\epsilon\text{ys})=(\text{A}'\text{lx}\epsilon\text{ys})$, 1 $\text{A}\langle\text{y}/\text{x}\rangle=\text{A}'\langle\text{y}/\text{x}\rangle$, 2 $\text{A}\langle\text{y}/\text{x}\rangle$, 3 $\text{A}'\langle\text{y}/\text{x}\rangle$, 4 $(\text{Alx}\epsilon\text{ys})$, 5 $(\text{A}'\text{lx}\epsilon\text{ys})$,
 6 $(\text{A}\langle\text{y}/\text{x}\rangle)\text{@}(\text{Alx}\epsilon\text{ys})=(\text{A}'\langle\text{y}/\text{x}\rangle)\text{@}(\text{A}'\text{lx}\epsilon\text{ys})$, 7 $\text{xx}4$, 8 $\text{A}\langle\text{y}/\text{x}\rangle=\text{A}'\langle\text{y}/\text{x}\rangle\wedge(\text{Alx}\epsilon\text{ys})=(\text{A}'\text{lx}\epsilon\text{ys})$, 9 A' , 10 y , 11 ys ,
 12 x , 13 $(\text{A}'\text{lx}\epsilon(\text{y})\text{@ys})$, 14 $\text{xx}3$, 15 $(\text{A}'\langle\text{y}/\text{x}\rangle)\text{@}(\text{A}'\text{lx}\epsilon\text{ys})$, 16 $(\text{A}\langle\text{y}/\text{x}\rangle)\text{@}(\text{Alx}\epsilon\text{ys})=\text{xx}3$, 17 A , 18 $(\text{Alx}\epsilon(\text{y})\text{@ys})$,
 19 $\text{xx}2$, 20 $(\text{A}\langle\text{y}/\text{x}\rangle)\text{@}(\text{Alx}\epsilon\text{ys})$, 21 $\text{xx}2=(\text{A}'\text{lx}\epsilon(\text{y})\text{@ys})$, 22 $(\text{Alx}\epsilon(\text{y})\text{@ys})=(\text{A}'\text{lx}\epsilon(\text{y})\text{@ys})$,
 23 $\forall\text{x}:(\text{x}\epsilon\text{set ys}\rightarrow\text{A}=\text{A}')\rightarrow(\text{Alx}\epsilon\text{ys})=(\text{A}'\text{lx}\epsilon\text{ys})$, 24 $\text{A}\langle\text{cc}/\text{x}\rangle=\text{A}'\langle\text{cc}/\text{x}\rangle$, 25 $\text{cc}\epsilon\text{set}(\text{y})\text{@ys}\rightarrow\text{A}\langle\text{cc}/\text{x}\rangle=\text{A}'\langle\text{cc}/\text{x}\rangle$,
 26 $\text{cc}\epsilon\text{set ys}$, 27 cc , 28 $\text{set}(\text{y})$, 29 set ys , 30 (y) , 31 $\text{set}(\text{y})\text{@ys}$, 32 $\text{xx}8$, 33 $\text{set}(\text{y})\text{uset ys}$, 34 $\text{cc}\epsilon\text{xx}8$,
 35 $\text{cc}\epsilon\text{set}(\text{y})\text{@ys}$, 36 $\forall\text{x}:(\text{x}\epsilon\text{set}(\text{y})\text{@ys}\rightarrow\text{A}=\text{A}')$, 37 $\text{x}\epsilon\text{set}(\text{y})\text{@ys}\rightarrow\text{A}=\text{A}'$, 38 $\text{x}\epsilon\text{set ys}\rightarrow\text{A}=\text{A}'$,
 39 $\forall\text{x}:(\text{x}\epsilon\text{set ys}\rightarrow\text{A}=\text{A}')$, 40 $\text{y}\epsilon\text{set}(\text{y})\text{@ys}\rightarrow\text{A}\langle\text{y}/\text{x}\rangle=\text{A}'\langle\text{y}/\text{x}\rangle$, 41 $\text{y}\epsilon\{\text{y}\}$, 42 $\text{xx}7$, 43 $\text{y}=\text{y}$, 44 $\{\text{y}\}$, 45 $\text{xx}6$,
 46 $\text{y}\epsilon\text{xx}6\text{uset ys}$, 47 $\text{xx}5$, 48 $\text{y}\epsilon\text{xx}5$, 49 $\text{y}\epsilon\text{set}(\text{y})\text{@ys}$, 50 $\forall\text{x}:(\text{x}\epsilon\text{set}(\text{y})\text{@ys}\rightarrow\text{A}=\text{A}')$, 51 $()$, 52 $(\text{A}'\text{lx}\epsilon())$,
 53 $\text{xx}1$, 54 $()=\text{xx}1$, 55 $(\text{Alx}\epsilon())$, 56 xx , 57 $\text{xx}=(\text{A}'\text{lx}\epsilon())$, 58 $\forall\text{x}:(\text{x}\epsilon\text{set}()\rightarrow\text{A}=\text{A}')$, 59 $(\text{Alx}\epsilon())=(\text{A}'\text{lx}\epsilon())$,
 60 $\forall\text{x}:(\text{x}\epsilon\text{set xs}\rightarrow\text{A}=\text{A}')\rightarrow(\text{Alx}\epsilon\text{xs})=(\text{A}'\text{lx}\epsilon\text{xs})$, 61 xs , 62 S ,
 63 finitesequence $\text{xs}\rightarrow\forall\text{x}:(\text{x}\epsilon\text{set xs}\rightarrow\text{A}=\text{A}')\rightarrow(\text{Alx}\epsilon\text{xs})=(\text{A}'\text{lx}\epsilon\text{xs})$, 64 finitesequence S ,
 65 $\forall\text{x}:(\text{x}\epsilon\text{set S}\rightarrow\text{A}=\text{A}')\rightarrow(\text{Alx}\epsilon\text{S})=(\text{A}'\text{lx}\epsilon\text{S})$, 66 $\forall\text{x}:(\text{x}\epsilon\text{set S}\rightarrow\text{A}=\text{A}')$, 67 $(\text{Alx}\epsilon\text{S})=(\text{A}'\text{lx}\epsilon\text{S})$

SEQ

```
(cut«64,67/B,C»)
(GIVEN 0)
(cut«66,67/B,C»)
(GIVEN 1)
("→-E"«66,67/A,B»)
(hyp«66/A»)
("→-E"«64,65/A,B»)
(hyp«64/A»)
("∀-E"«62,63,61/B,A,x»)
("finite sequence induction (L)"«10,11,60,61/y,ys,P,x»)
("→-I"«58,59/A,B»)
(LAYOUT "(Alxε())≐()" ALL
  ("rewrite≐"«55,56,51,57/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("(Alxε())≐()"«17,12/A,x»)
  (LAYOUT "(Alxε())≐()" ALL
    ("rewrite≐"«52,53,51,54/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("(Alxε())≐()"«9,12/A,x»)
    (LAYOUT HIDEROOT
      ("reflexive="«51/A»))))
("→-I"«50,22/A,B»)
(cut«40,22/B,C»)
("∀-E"«10,37,12/B,A,x»)
(hyp«36/A»)
(cut«1,22/B,C»)
("→-E"«49,1/A,B»)
(LAYOUT "set(R@S)≐set Ruset S" ALL
  ("rewrite≐"«31,47,33,48/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("set(R@S)≐set Ruset S"«30,11/R,S»)
  (LAYOUT "set(A)≐{A}" ALL
    ("rewrite≐"«28,45,44,46/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("set(A)≐{A}"«10/A»)
    ("FROM A∈S INFER A∈SUT"«10,44,29/A,S,T»)
    (LAYOUT "A∈{B}≐A=B" ALL
      ("rewrite≐"«41,42,43,42/A,xx,B,P»)
      (LAYOUT HIDEROOT
        ("A∈{B}≐A=B"«10,10/A,B»)
      (LAYOUT HIDEROOT
        ("reflexive="«10/A»))))))
(hyp«40/A»)
(cut«1,22/B,C»)
(hyp«1/A»)
(cut«0,22/B,C»)
("→-E"«39,0/A,B»)
("∀-I"«27,38,12/cc,A,x»)
("→-I"«26,24/A,B»)
(cut«25,24/B,C»)
("∀-E"«27,37,12/B,A,x»)
```

```

(hyp«36/A»)
(cut«24,24/B,C»)
("→-E"«35,24/A,B»)
(LAYOUT "set(R@S)≐set R∪set S" ALL
  ("rewrite≐"«31,32,33,34/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("set(R@S)≐set R∪set S"«30,11/R,S»)
    ("FROM A∈T INFER A∈S∪T"«27,28,29/A,S,T»)
    (hyp«26/A»))
(hyp«25/A»)
(hyp«24/A»)
(hyp«23/A»)
(cut«0,22/B,C»)
(hyp«0/A»)
(LAYOUT "{A|x∈(B)@S}≐{A}«B/x»@{A|x∈S}" ALL
  ("rewrite≐"«18,19,20,21/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A|x∈(B)@S}≐{A}«B/x»@{A|x∈S}"«17,10,11,12/A,B,S,x»))
(LAYOUT "{A|x∈(B)@S}≐{A}«B/x»@{A|x∈S}" ALL
  ("rewrite≐"«13,14,15,16/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A|x∈(B)@S}≐{A}«B/x»@{A|x∈S}"«9,10,11,12/A,B,S,x»))
(LAYOUT "{A}@R=(B)@S≐A=B∧R=S" ALL
  ("rewrite≐"«6,7,8,7/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A}@R=(B)@S≐A=B∧R=S"«2,3,4,5/A,B,R,S»))
(LAYOUT COMPRESS "∧-I" ALL
  ("∧-I"«1,0/A,B»)
  (hyp«1/A»)
  (hyp«0/A»))))

```

1.3.3.5. Ordered lists

DERIVED RULE IS FROM olist(E,S) INFER list S

olist(E,S) 1: olist(E,S)

(1) olist(E,S)≐oseq(E,S)∧list S 2: oseq(E,S)∧list S

(2) ∧-E 3: list S

Given:

olist(E,S)

FORMULAE 0 list S, 1 oseq(E,S)∧list S, 2 oseq(E,S), 3 olist(E,S), 4 S, 5 E, 6 xx

SEQ

```

(cut«3,0/B,C»)
(GIVEN 0)
(cut«1,0/B,C»)
(LAYOUT "olist(E,S)≐oseq(E,S)∧list S" ALL
  ("rewrite≐"«1,6,3,6/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«3,1/A,B»)
    (LAYOUT HIDEROOT
      ("olist(E,S)≐oseq(E,S)∧list S"«4,5/S,E»)))
  (hyp«3/A»))
(cut«2,0/B,C»)
(LAYOUT "∧-E" ALL
  ("∧-E(L)"«0,2/B,A»)
  (hyp«1/A»))
(cut«0,0/B,C»)
(LAYOUT "∧-E" ALL
  ("∧-E(R)"«2,0/A,B»)

```

(hyp«1/A»)
(hyp«0/A»)

DERIVED RULE IS FROM olist(E,S) INFER oseq(E,S)

olist(E,S) 1: olist(E,S)

(1) olist(E,S)≐oseq(E,S)∧list S 2: oseq(E,S)∧list S

(2) ∧-E 3: oseq(E,S)

Given:

olist(E,S)

FORMULAE 0 oseq(E,S), 1 oseq(E,S)∧list S, 2 list S, 3 olist(E,S), 4 S, 5 E, 6 xx

SEQ

(cut«3,0/B,C»)
(GIVEN 0)
(cut«1,0/B,C»)
(LAYOUT "olist(E,S)≐oseq(E,S)∧list S" ALL
("rewrite≐"«1,6,3,6/A,xx,B,P»)
(LAYOUT HIDEROOT
("symmetric≐"«3,1/A,B»)
(LAYOUT HIDEROOT
("olist(E,S)≐oseq(E,S)∧list S"«4,5/S,E»)))
(hyp«3/A»)
(cut«0,0/B,C»)
(LAYOUT "∧-E" ALL
("∧-E(L)"«2,0/B,A»)
(hyp«1/A»)
(cut«2,0/B,C»)
(LAYOUT "∧-E" ALL
("∧-E(R)"«0,2/A,B»)
(hyp«1/A»)
(hyp«0/A»)

DERIVED RULE IS FROM olist(E,R@S) INFER olist(E,R) ∧ olist(E,S)

No proof: can't recall why; don't know if it's used anywhere. Doesn't look hard. (Or is it a very tedious induction?)

THEOREM IS olist(E,())

oseq(E,()) 1: oseq(E,())

(1) ∧-I 2: oseq(E,())∧list()

(2) olist(E,S)≐oseq(E,S)∧list S 3: olist(E,())

FORMULAE 0 list(), 1 oseq(E,()), 2 E, 3 (), 4 olist(E,()), 5 xx, 6 oseq(E,())∧list()

SEQ

(cut«1,4/B,C»)
("oseq(E,())"«2/E»)
(cut«0,4/B,C»)
(LAYOUT HIDEROOT
("list()"))
(LAYOUT "olist(E,S)≐oseq(E,S)∧list S" ALL
("rewrite≐"«4,5,6,5/A,xx,B,P»)
(LAYOUT HIDEROOT
("olist(E,S)≐oseq(E,S)∧list S"«2,3/E,S»))
(LAYOUT COMPRESS "∧-I" ALL
("∧-I"«1,0/A,B»)

(hyp«1/A»
(hyp«0/A»)))

THEOREM IS olist(E,(A))

oseq(E,(A)) 1: oseq(E,(A))
(1) \wedge -I 2: oseq(E,(A)) \wedge list(A)

(2) olist(E,S) $\hat{=}$ oseq(E,S) \wedge list S 3: olist(E,(A))

0 list(A), 1 oseq(E,(A)), 2 E, 3 (A), 4 olist(E,(A)), 5 xx, 6 oseq(E,(A)) \wedge list(A), 7 A

SEQ

(cut«1,4/B,C»)
("oseq(E,(A))"«7,2/A,E»)
(cut«0,4/B,C»)
(LAYOUT HIDEROOT
("list(A)"«7/A»))
(LAYOUT "olist(E,S) $\hat{=}$ oseq(E,S) \wedge list S" ALL
("rewrite"«4,5,6,5/A,xx,B,P»)
(LAYOUT HIDEROOT
("olist(E,S) $\hat{=}$ oseq(E,S) \wedge list S"«2,3/E,S»))
(LAYOUT COMPRESS " \wedge -I" ALL
(" \wedge -I"«1,0/A,B»)
(hyp«1/A»
(hyp«0/A»)))

1.3.3.6. Sets

DERIVED RULE IS $\{\} \subseteq S$

1:	$cc \in \{\}$	assumption
2:	$\neg(cc \in \{\})$	$\neg(A \in \{\})$
3:	\perp	\neg -E 1,2
4:	$cc \in S$	\perp -E 3
5:	$cc \in \{\} \rightarrow cc \in S$	\rightarrow -I 1-4
6:	$\forall x:(x \in \{\} \rightarrow x \in S)$	\forall -I 5
7:	$\{\} \subseteq S$	$A \subseteq B \hat{=} \forall x:(x \in A \rightarrow x \in B)$ 6

FORMULAE 0 cc, 1 $cc \in \{\}$, 2 $cc \in S$, 3 $x \in \{\} \rightarrow x \in S$, 4 x, 5 $\{\}$, 6 S, 7 $\{\} \subseteq S$, 8 xx, 9 $\forall x:(x \in \{\} \rightarrow x \in S)$

LAYOUT "A \subseteq B $\hat{=} \forall x:(x \in A \rightarrow x \in B)"$ ALL
("rewrite"«7,8,9,8/A,xx,B,P»)
(LAYOUT HIDEROOT
("A \subseteq B $\hat{=} \forall x:(x \in A \rightarrow x \in B)"$ «4,5,6/x,A,B»))
(" \forall -I"«0,3,4/cc,A,x»)
(" \rightarrow -I"«1,2/A,B»)
(" \perp -E"«2/A»)
(" \neg -E"«1/B»)
(hyp«1/A»
(" $\neg(A \in \{\})"$ «0/A»))

DERIVED RULE IS $S \subseteq S$

- 1: $\boxed{cc \in S}$ assumption
- 2: $cc \in S \rightarrow cc \in S$ \rightarrow -I 1-1
- 3: $\forall x: (x \in S \rightarrow x \in S)$ \forall -I 2
- 4: $S \subseteq S$ $A \subseteq B \triangleq \forall x: (x \in A \rightarrow x \in B)$ 3

FORMULAE 0 $cc \in S$, 1 cc , 2 $x \in S \rightarrow x \in S$, 3 x , 4 S , 5 $S \subseteq S$, 6 xx , 7 $\forall x: (x \in S \rightarrow x \in S)$

LAYOUT "A \subseteq B \triangleq $\forall x: (x \in A \rightarrow x \in B)$ " ALL
 ("rewrite" \triangleq "«5,6,7,6/A,xx,B,P»")
 (LAYOUT HIDEROOT
 ("A \subseteq B \triangleq $\forall x: (x \in A \rightarrow x \in B)$ " «3,4,4/x,A,B»))
 ("V-I" «1,2,3/cc,A,x»)
 ("→-I" «0,0/A,B»)
 (hyp«0/A»)

DERIVED RULE (OBJECT cx) WHERE FRESH cx IS FROM $cx \in S \vdash cx \in T$ INFER $S \subseteq T$

- 1: $\boxed{cx \in S}$ assumption
- 2: $\boxed{cx \in T}$ $cx \in S \vdash cx \in T$ 1
- 3: $cx \in S \rightarrow cx \in T$ \rightarrow -I 1-2
- 4: $\forall x: (x \in S \rightarrow x \in T)$ \forall -I 3
- 5: $S \subseteq T$ $A \subseteq B \triangleq \forall x: (x \in A \rightarrow x \in B)$ 4

Given:

$cx \in S \vdash cx \in T$

Provided:

FRESH cx

FORMULAE 0 $cx \in S$, 1 $cx \in T$, 2 cx , 3 $x \in S \rightarrow x \in T$, 4 x , 5 S , 6 T , 7 $S \subseteq T$, 8 xx , 9 $\forall x: (x \in S \rightarrow x \in T)$

LAYOUT "A \subseteq B \triangleq $\forall x: (x \in A \rightarrow x \in B)$ " ALL
 ("rewrite" \triangleq "«7,8,9,8/A,xx,B,P»")
 (LAYOUT HIDEROOT
 ("A \subseteq B \triangleq $\forall x: (x \in A \rightarrow x \in B)$ " «4,5,6/x,A,B»))
 ("V-I" «2,3,4/cc,A,x»)
 ("→-I" «0,1/A,B»)
 (GIVEN 0)

DERIVED RULE IS FROM $S \subseteq T$ INFER $A \in S \rightarrow A \in T$

- 1: $S \subseteq T$ $S \subseteq T$
- 2: $\forall x: (x \in S \rightarrow x \in T)$ $A \subseteq B \triangleq \forall x: (x \in A \rightarrow x \in B)$ 1
- 3: $A \in S \rightarrow A \in T$ \forall -E 2

Given:

$S \subseteq T$

FORMULAE 0 $A \in S \rightarrow A \in T$, 1 $\forall x: (x \in S \rightarrow x \in T)$, 2 A , 3 $x \in S \rightarrow x \in T$, 4 x , 5 $S \subseteq T$, 6 S , 7 T , 8 $\forall x: (x \in S \rightarrow x \in T)$, 9 xx

SEQ

(cut«5,0/B,C»)

(GIVEN 0)

(cut«8,0/B,C»)

(LAYOUT "A \subseteq B \triangleq $\forall x: (x \in A \rightarrow x \in B)$ " ALL

("rewrite" \triangleq "«8,9,5,9/A,xx,B,P»")

(LAYOUT HIDEROOT
("symmetric $\hat{=}$ " $\langle 5, 8/A, B \rangle$)
(LAYOUT HIDEROOT
("A \in B $\hat{=}$ $\forall x:(x \in A \rightarrow x \in B)$ " $\langle 4, 6, 7/x, A, B \rangle$)))
(hyp $\langle 5/A \rangle$)
(cut $\langle 0, 0/B, C \rangle$)
("V-E" $\langle 2, 3, 4/B, A, x \rangle$)
(hyp $\langle 1/A \rangle$)
(hyp $\langle 0/A \rangle$)

DERIVED RULE IS $A \cup B \cup C = A \cup C \cup B$

1: $cx \in A \cup B \cup C$	assumption
2: $cx \in A \cup B \vee cx \in C$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 1
3: $cx \in A \cup B$	assumption
4: $cx \in A \vee cx \in B$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 3
5: $cx \in A$	assumption
6: $cx \in A \cup C$	Derived Rule FROM $A \in S$ INFER $A \in S \cup T$ 5
7: $cx \in A \cup C \vee cx \in B$	\vee -I(L) 6
8: $cx \in A \cup C \cup B$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 7
9: $cx \in B$	assumption
10: $cx \in A \cup C \cup B$	Derived Rule FROM $A \in T$ INFER $A \in S \cup T$ 9
11: $cx \in A \cup C \cup B$	\vee -E 4,5-8,9-10
12: $cx \in C$	assumption
13: $cx \in A \cup C$	Derived Rule FROM $A \in T$ INFER $A \in S \cup T$ 12
14: $cx \in A \cup C \vee cx \in B$	\vee -I(L) 13
15: $cx \in A \cup C \cup B$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 14
16: $cx \in A \cup C \cup B$	\vee -E 2,3-11,12-15
17: $A \cup B \cup C \subseteq A \cup C \cup B$	Derived Rule FROM $cx \in S \vdash cx \in T$ INFER $S \subseteq T$ 1-16
18: $cx1 \in A \cup C \cup B$	assumption
19: $cx1 \in A \cup C \vee cx1 \in B$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 18
20: $cx1 \in A \cup C$	assumption
21: $cx1 \in A \vee cx1 \in C$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 20
22: $cx1 \in A$	assumption
23: $cx1 \in A \cup B$	Derived Rule FROM $A \in S$ INFER $A \in S \cup T$ 22
24: $cx1 \in A \cup B \vee cx1 \in C$	\vee -I(L) 23
25: $cx1 \in A \cup B \cup C$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 24
26: $cx1 \in C$	assumption
27: $cx1 \in A \cup B \cup C$	Derived Rule FROM $A \in T$ INFER $A \in S \cup T$ 26
28: $cx1 \in A \cup B \cup C$	\vee -E 21,22-25,26-27
29: $cx1 \in B$	assumption
30: $cx1 \in A \cup B$	Derived Rule FROM $A \in T$ INFER $A \in S \cup T$ 29
31: $cx1 \in A \cup B \vee cx1 \in C$	\vee -I(L) 30
32: $cx1 \in A \cup B \cup C$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 31
33: $cx1 \in A \cup B \cup C$	\vee -E 19,20-28,29-32
34: $A \cup C \cup B \subseteq A \cup B \cup C$	Derived Rule FROM $cx \in S \vdash cx \in T$ INFER $S \subseteq T$ 18-33
35: $A \cup B \cup C \subseteq A \cup C \cup B \wedge A \cup C \cup B \subseteq A \cup B \cup C$	\wedge -I 17,34
36: $A \cup B \cup C = A \cup C \cup B$	$S = T \triangleq S \subseteq T \wedge T \subseteq S$ 35

FORMULAE 0 $A \cup C \cup B \subseteq A \cup B \cup C$, 1 $A \cup B \cup C \subseteq A \cup C \cup B$, 2 $A \cup B \cup C$, 3 $A \cup C \cup B$, 4 $A \cup B \cup C = A \cup C \cup B$, 5 xx ,
 6 $A \cup B \cup C \subseteq A \cup C \cup B \wedge A \cup C \cup B \subseteq A \cup B \cup C$, 7 $cx1 \in B$, 8 $cx1$, 9 A , 10 B , 11 $cx1 \in C$, 12 $cx1 \in A \cup B$, 13 $A \cup B$, 14 C ,
 15 $cx1 \in A \cup B \cup C$, 16 $xx8$, 17 $cx1 \in A \cup B \vee cx1 \in C$, 18 $cx1 \in A$, 19 $xx7$, 20 $cx1 \in A \vee cx1 \in C$, 21 $cx1 \in A \cup C$, 22 $xx6$,
 23 $cx1 \in A \cup C \vee cx1 \in B$, 24 $cx1 \in A \cup C \cup B$, 25 $A \cup C$, 26 $xx5$, 27 $cx \in C$, 28 cx , 29 $cx \in B$, 30 $cx \in A \cup C$, 31 $cx \in A \cup C \cup B$,
 32 $xx4$, 33 $cx \in A \cup C \vee cx \in B$, 34 $cx \in A$, 35 $xx3$, 36 $cx \in A \vee cx \in B$, 37 $cx \in A \cup B$, 38 $xx2$, 39 $cx \in A \cup B \vee cx \in C$,
 40 $cx \in A \cup B \cup C$, 41 $xx1$

SEQ

```
(cut«1,4/B,C»)
("FROM  $cx \in S \vdash cx \in T$  INFER  $S \subseteq T$ "«28,2,3/cx,S,T»)
(cut«39,31/B,C»)
(LAYOUT "A  $\in R \cup S \supseteq A \in R \vee A \in S$ " ALL
  ("rewrite $\hat{=}$ "«39,41,40,41/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric $\hat{=}$ "«40,39/A,B»)
    (LAYOUT HIDEROOT
      ("A  $\in R \cup S \supseteq A \in R \vee A \in S$ "«28,13,14/A,R,S»)))
  (hyp«40/A»)
("  $\vee$ -E"«37,27,31/A,B,C»)
(hyp«39/A»)
(cut«36,31/B,C»)
(LAYOUT "A  $\in R \cup S \supseteq A \in R \vee A \in S$ " ALL
  ("rewrite $\hat{=}$ "«36,38,37,38/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric $\hat{=}$ "«37,36/A,B»)
    (LAYOUT HIDEROOT
      ("A  $\in R \cup S \supseteq A \in R \vee A \in S$ "«28,9,10/A,R,S»)))
  (hyp«37/A»)
("  $\vee$ -E"«34,29,31/A,B,C»)
(hyp«36/A»)
(LAYOUT "A  $\in R \cup S \supseteq A \in R \vee A \in S$ " ALL
  ("rewrite $\hat{=}$ "«31,35,33,35/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A  $\in R \cup S \supseteq A \in R \vee A \in S$ "«28,25,10/A,R,S»)
    ("  $\vee$ -I(L)"«29,30/B,A»)
    ("FROM  $A \in S$  INFER  $A \in S \cup T$ "«28,9,14/A,S,T»)
    (hyp«34/A»)
    ("FROM  $A \in T$  INFER  $A \in S \cup T$ "«28,25,10/A,S,T»)
    (hyp«29/A»)
    (LAYOUT "A  $\in R \cup S \supseteq A \in R \vee A \in S$ " ALL
      ("rewrite $\hat{=}$ "«31,32,33,32/A,xx,B,P»)
      (LAYOUT HIDEROOT
        ("A  $\in R \cup S \supseteq A \in R \vee A \in S$ "«28,25,10/A,R,S»)
        ("  $\vee$ -I(L)"«29,30/B,A»)
        ("FROM  $A \in T$  INFER  $A \in S \cup T$ "«28,9,14/A,S,T»)
        (hyp«27/A»)
      )
    )
  )
(cut«0,4/B,C»)
("FROM  $cx \in S \vdash cx \in T$  INFER  $S \subseteq T$ "«8,3,2/cx,S,T»)
(cut«23,15/B,C»)
(LAYOUT "A  $\in R \cup S \supseteq A \in R \vee A \in S$ " ALL
  ("rewrite $\hat{=}$ "«23,26,24,26/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric $\hat{=}$ "«24,23/A,B»)
    (LAYOUT HIDEROOT
      ("A  $\in R \cup S \supseteq A \in R \vee A \in S$ "«8,25,10/A,R,S»)))
  (hyp«24/A»)
("  $\vee$ -E"«21,7,15/A,B,C»)
(hyp«23/A»)
(cut«20,15/B,C»)
(LAYOUT "A  $\in R \cup S \supseteq A \in R \vee A \in S$ " ALL
  ("rewrite $\hat{=}$ "«20,22,21,22/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric $\hat{=}$ "«21,20/A,B»)
    (LAYOUT HIDEROOT
      ("A  $\in R \cup S \supseteq A \in R \vee A \in S$ "«8,9,14/A,R,S»)))
  (hyp«21/A»)
("  $\vee$ -E"«18,11,15/A,B,C»)
```

```

(hyp«20/A»)
(LAYOUT "A∈RUS≐A∈R∨A∈S" ALL
  ("rewrite≐"«15,19,17,19/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A∈RUS≐A∈R∨A∈S"«8,13,14/A,R,S»))
  ("∨-I(L)"«11,12/B,A»)
  ("FROM A∈S INFER A∈SUT"«8,9,10/A,S,T»)
  (hyp«18/A»)
  ("FROM A∈T INFER A∈SUT"«8,13,14/A,S,T»)
  (hyp«11/A»)
(LAYOUT "A∈RUS≐A∈R∨A∈S" ALL
  ("rewrite≐"«15,16,17,16/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A∈RUS≐A∈R∨A∈S"«8,13,14/A,R,S»))
  ("∨-I(L)"«11,12/B,A»)
  ("FROM A∈T INFER A∈SUT"«8,9,10/A,S,T»)
  (hyp«7/A»)
(LAYOUT "S=T≐S⊆T∧T⊆S" ALL
  ("rewrite≐"«4,5,6,5/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("S=T≐S⊆T∧T⊆S"«2,3/S,T»))
(LAYOUT COMPRESS "∧-I" ALL
  ("∧-I"«1,0/A,B»)
  (hyp«1/A»)
  (hyp«0/A»)))

```

DERIVED RULE IS FROM $A \in S$ INFER $A \in S_U T$

- 1: $A \in S$ $A \in S$
- 2: $A \in S_{\vee} A \in T$ $\vee\text{-I(L)} \ 1$
- 3: $A \in S_U T$ $A \in R_U S \doteq A \in R_{\vee} A \in S \ 2$

Given:
 $A \in S$

FORMULAE 0 $A \in T$, 1 $A \in S$, 2 A , 3 S , 4 T , 5 $A \in S_U T$, 6 xx , 7 $A \in S_{\vee} A \in T$

```

LAYOUT "A∈RUS≐A∈R∨A∈S" ALL
  ("rewrite≐"«5,6,7,6/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A∈RUS≐A∈R∨A∈S"«2,3,4/A,R,S»))
  ("∨-I(L)"«0,1/B,A»)
  (GIVEN 0)

```

DERIVED RULE IS FROM $A \in T$ INFER $A \in S_U T$

- 1: $A \in T$ $A \in T$
- 2: $A \in S_{\vee} A \in T$ $\vee\text{-I(R)} \ 1$
- 3: $A \in S_U T$ $A \in R_U S \doteq A \in R_{\vee} A \in S \ 2$

Given:
 $A \in T$

FORMULAE 0 $A \in S$, 1 $A \in T$, 2 A , 3 S , 4 T , 5 $A \in S_U T$, 6 xx , 7 $A \in S_{\vee} A \in T$

```

LAYOUT "A∈RUS≐A∈R∨A∈S" ALL
  ("rewrite≐"«5,6,7,6/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A∈RUS≐A∈R∨A∈S"«2,3,4/A,R,S»))

```

(\neg -I(R)«0,1/A,B») (GIVEN 0)

DERIVED RULE IS FROM $\neg(A \in S)$ AND $\neg(A \in T)$ INFER $\neg(A \in S \cup T)$

1: $A \in S \cup T$	assumption
2: $A \in S \vee A \in T$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 1
3: $A \in S$	assumption
4: $\neg(A \in S)$	$\neg(A \in S)$
5: \perp	\neg -E 3,4
6: $A \in T$	assumption
7: $\neg(A \in T)$	$\neg(A \in T)$
8: \perp	\neg -E 6,7
9: \perp	\vee -E 2,3-5,6-8
10: $\neg(A \in S \cup T)$	\neg -I 1-9

Given:
 $\neg(A \in S)$
 $\neg(A \in T)$

FORMULAE 0 $A \in T$, 1 $A \in S$, 2 $A \in S \vee A \in T$, 3 \perp , 4 $A \in S \cup T$, 5 A, 6 S, 7 T, 8 xx

SEQ
 (\neg -I«4/A») (cut«2,3/B,C») (LAYOUT "A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S" ALL ("rewrite≐"«2,8,4,8/A,xx,B,P») (LAYOUT HIDEROOT ("symmetric≐"«4,2/A,B») (LAYOUT HIDEROOT ("A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S"«5,6,7/A,R,S»))) (hyp«4/A») (\neg -E«1,0,3/A,B,C») (hyp«2/A») (\neg -E«1/B») (hyp«1/A») (GIVEN 0) (\neg -E«0/B») (hyp«0/A») (GIVEN 1)

DERIVED RULE IS FROM $R \subseteq S$ INFER $R \subseteq S \cup T$

1: $R \subseteq S$	$R \subseteq S$
2: $cc \in R$	assumption
3: $cc \in R \rightarrow cc \in S$	Derived Rule FROM $S \subseteq T$ INFER $A \in S \rightarrow A \in T$ 1
4: $cc \in S$	\rightarrow -E 2,3
5: $cc \in S \vee cc \in T$	\vee -I(L) 4
6: $cc \in S \cup T$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 5
7: $cc \in R \rightarrow cc \in S \cup T$	\rightarrow -I 2-6
8: $\forall x: (x \in R \rightarrow x \in S \cup T)$	\forall -I 7
9: $R \subseteq S \cup T$	$A \subseteq B \triangleq \forall x: (x \in A \rightarrow x \in B)$ 8

Given:

$R \subseteq S$

FORMULAE 0 $cc \in S$, 1 $cc \in T$, 2 cc , 3 S , 4 T , 5 $cc \in S \cup T$, 6 xx , 7 $cc \in S \vee cc \in T$, 8 $cc \in R \rightarrow cc \in S$, 9 $cc \in R$, 10 $R \subseteq S$, 11 R , 12 $x \in R \rightarrow x \in S \cup T$, 13 x , 14 $S \cup T$, 15 $R \subseteq S \cup T$, 16 xx , 17 $\forall x: (x \in R \rightarrow x \in S \cup T)$

SEQ

```
(cut«10,15/B,C»)
(GIVEN 0)
(LAYOUT "A ⊆ B ≐ ∀x: (x ∈ A → x ∈ B)" ALL
  ("rewrite≐"«15,16,17,16/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A ⊆ B ≐ ∀x: (x ∈ A → x ∈ B)"«13,11,14/x,A,B»)
    ("∀-I"«2,12,13/cc,A,x»)
    ("→-I"«9,5/A,B»)
    (cut«8,5/B,C»)
    ("FROM S ⊆ T INFER A ∈ S → A ∈ T"«2,11,3/A,S,T»)
    (hyp«10/A»)
    (cut«0,5/B,C»)
    ("→-E"«9,0/A,B»)
    (hyp«9/A»)
    (hyp«8/A»)
  (LAYOUT "A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S" ALL
    ("rewrite≐"«5,6,7,6/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S"«2,3,4/A,R,S»)
      ("∨-I(L)"«1,0/B,A»)
      (hyp«0/A»)))
  (hyp«0/A»)))
```

DERIVED RULE IS FROM $R \subseteq T$ INFER $R \subseteq S \cup T$

1: $R \subseteq T$	$R \subseteq T$
2: $cc \in R$	assumption
3: $cc \in R \rightarrow cc \in T$	Derived Rule FROM $S \subseteq T$ INFER $A \in S \rightarrow A \in T$ 1
4: $cc \in T$	\rightarrow -E 2,3
5: $cc \in S \vee cc \in T$	\vee -I(R) 4
6: $cc \in S \cup T$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 5
7: $cc \in R \rightarrow cc \in S \cup T$	\rightarrow -I 2-6
8: $\forall x: (x \in R \rightarrow x \in S \cup T)$	\forall -I 7
9: $R \subseteq S \cup T$	$A \subseteq B \triangleq \forall x: (x \in A \rightarrow x \in B)$ 8

Given:

$R \subseteq T$

FORMULAE 0 $cc \in T$, 1 $cc \in S$, 2 cc , 3 S , 4 T , 5 $cc \in S \cup T$, 6 xx , 7 $cc \in S \vee cc \in T$, 8 $cc \in R \rightarrow cc \in T$, 9 $cc \in R$, 10 $R \subseteq T$, 11 R , 12 $x \in R \rightarrow x \in S \cup T$, 13 x , 14 $S \cup T$, 15 $R \subseteq S \cup T$, 16 xx , 17 $\forall x: (x \in R \rightarrow x \in S \cup T)$

SEQ

```
(cut«10,15/B,C»)
(GIVEN 0)
(LAYOUT "A ⊆ B ≐ ∀x: (x ∈ A → x ∈ B)" ALL
  ("rewrite≐"«15,16,17,16/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("A ⊆ B ≐ ∀x: (x ∈ A → x ∈ B)"«13,11,14/x,A,B»)
    ("∀-I"«2,12,13/cc,A,x»)
    ("→-I"«9,5/A,B»)
    (cut«8,5/B,C»)
    ("FROM S ⊆ T INFER A ∈ S → A ∈ T"«2,11,4/A,S,T»)
    (hyp«10/A»)
    (cut«0,5/B,C»)
    ("→-E"«9,0/A,B»)
    (hyp«9/A»)
    (hyp«8/A»)
  (LAYOUT "A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S" ALL
    ("rewrite≐"«5,6,7,6/A,xx,B,P»)
    (LAYOUT HIDEROOT
      ("A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S"«2,3,4/A,R,S»)
      ("∨-I(R)"«1,0/A,B»)
      (hyp«0/A»)))
  (hyp«0/A»)))
```

DERIVED RULE IS FROM $S1 \subseteq S2$ AND $T1 \subseteq T2$ INFER $S1 \cup T1 \subseteq S2 \cup T2$

1: $S1 \subseteq S2$	$S1 \subseteq S2$
2: $T1 \subseteq T2$	$T1 \subseteq T2$
3: $CX \in S1 \cup T1$	assumption
4: $CX \in S1 \vee CX \in T1$	$A \in R \cup S \triangleq A \in R \vee A \in S$ 3
5: $CX \in S1$	assumption
6: $CX \in S1 \rightarrow CX \in S2$	Derived Rule FROM $S \subseteq T$ INFER $A \in S \rightarrow A \in T$ 1
7: $CX \in S2$	\rightarrow -E 5,6
8: $CX \in S2 \cup T2$	Derived Rule FROM $A \in S$ INFER $A \in S \cup T$ 7
9: $CX \in T1$	assumption
10: $CX \in T1 \rightarrow CX \in T2$	Derived Rule FROM $S \subseteq T$ INFER $A \in S \rightarrow A \in T$ 2
11: $CX \in T2$	\rightarrow -E 9,10
12: $CX \in S2 \cup T2$	Derived Rule FROM $A \in T$ INFER $A \in S \cup T$ 11
13: $CX \in S2 \cup T2$	\vee -E 4,5-8,9-12
14: $S1 \cup T1 \subseteq S2 \cup T2$	Derived Rule FROM $CX \in S \vdash CX \in T$ INFER $S \subseteq T$ 3-13

Given:

$S1 \subseteq S2$

$T1 \subseteq T2$

FORMULAE 0 $CX \in T2$, 1 CX , 2 $S2$, 3 $T2$, 4 $CX \in T1 \rightarrow CX \in T2$, 5 $CX \in T1$, 6 $CX \in S2 \cup T2$, 7 $T1 \subseteq T2$, 8 $T1$, 9 $CX \in S2$, 10 $CX \in S1 \rightarrow CX \in S2$, 11 $CX \in S1$, 12 $S1 \subseteq S2$, 13 $S1$, 14 $CX \in S1 \vee CX \in T1$, 15 $CX \in S1 \cup T1$, 16 xx , 17 $S1 \cup T1$, 18 $S2 \cup T2$, 19 $S1 \cup T1 \subseteq S2 \cup T2$

SEQ

```
(cut«12,19/B,C»)
(GIVEN 0)
(cut«7,19/B,C»)
(GIVEN 1)
("FROM  $CX \in S \vdash CX \in T$  INFER  $S \subseteq T$ "«1,17,18/ $CX,S,T$ )
(cut«14,6/B,C»)
(LAYOUT "A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S" ALL
  ("rewrite≐"«14,16,15,16/A,xx,B,P»)
  (LAYOUT HIDEROOT
    ("symmetric≐"«15,14/A,B»)
    (LAYOUT HIDEROOT
      ("A ∈ R ∪ S ≐ A ∈ R ∨ A ∈ S"«1,13,8/A,R,S»)))
  (hyp«15/A»)
  ("∨-E"«11,5,6/A,B,C»)
  (hyp«14/A»)
  (cut«10,6/B,C»)
  ("FROM  $S \subseteq T$  INFER  $A \in S \rightarrow A \in T$ "«1,13,2/A,S,T»)
  (hyp«12/A»)
  (cut«9,6/B,C»)
  ("→-E"«11,9/A,B»)
  (hyp«11/A»)
  (hyp«10/A»)
  ("FROM  $A \in S$  INFER  $A \in S \cup T$ "«1,2,3/A,S,T»)
  (hyp«9/A»)
  (cut«4,6/B,C»)
  ("FROM  $S \subseteq T$  INFER  $A \in S \rightarrow A \in T$ "«1,8,3/A,S,T»)
  (hyp«7/A»)
  (cut«0,6/B,C»)
  ("→-E"«5,0/A,B»)
  (hyp«5/A»)
```

(hyp«4/A»)
 ("FROM $A \in T$ INFER $A \in S_{\cup T}$ "«1,2,3/A,S,T»)
 (hyp«0/A»)

 DERIVED RULE IS FROM $R \subseteq S$ INFER $R_{\cup S} = S$

1: $R \subseteq S$	$R \subseteq S$
2: $cx \in R_{\cup S}$	assumption
3: $cx \in R \vee cx \in S$	$A \in R_{\cup S} \triangleq A \in R \vee A \in S$ 2
4: $cx \in R$	assumption
5: $cx \in R \rightarrow cx \in S$	Derived Rule FROM $S \subseteq T$ INFER $A \in S \rightarrow A \in T$ 1
6: $cx \in S$	\rightarrow -E 4,5
7: $cx \in S$	assumption
8: $cx \in S$	\vee -E 3,4-6,7-7
9: $R_{\cup S} \subseteq S$	Derived Rule FROM $cx \in S \vdash cx \in T$ INFER $S \subseteq T$ 2-8
10: $cx1 \in S$	assumption
11: $cx1 \in S_{\cup R}$	Derived Rule FROM $A \in S$ INFER $A \in S_{\cup T}$ 10
12: $S \subseteq S_{\cup R}$	Derived Rule FROM $cx \in S \vdash cx \in T$ INFER $S \subseteq T$ 10-11
13: $S \subseteq R_{\cup S}$	$R_{\cup S} \triangleq S_{\cup R}$ 12
14: $R_{\cup S} \subseteq S \wedge S \subseteq R_{\cup S}$	\wedge -I 9,13
15: $R_{\cup S} = S$	$S = T \triangleq S \subseteq T \wedge T \subseteq S$ 14

Given:
 $R \subseteq S$

FORMULAE 0 $S \subseteq R_{\cup S}$, 1 $R_{\cup S} \subseteq S$, 2 $R_{\cup S}$, 3 S , 4 $R_{\cup S} = S$, 5 xx , 6 $R_{\cup S} \subseteq S \wedge S \subseteq R_{\cup S}$, 7 $cx1 \in S$, 8 $cx1$, 9 R , 10 $S_{\cup R}$, 11 $xx2$, 12 $S \subseteq xx2$, 13 $cx \in S$, 14 $cx \in R \rightarrow cx \in S$, 15 $cx \in R$, 16 $R \subseteq S$, 17 cx , 18 $cx \in R \vee cx \in S$, 19 $cx \in R_{\cup S}$, 20 $xx1$

SEQ

(cut«16,4/B,C»)
 (GIVEN 0)
 (cut«1,4/B,C»)
 ("FROM $cx \in S \vdash cx \in T$ INFER $S \subseteq T$ "«17,2,3/cx,S,T»)
 (cut«18,13/B,C»)
 (LAYOUT "A $\in R_{\cup S} \triangleq A \in R \vee A \in S$ " ALL
 ("rewrite"«18,20,19,20/A,xx,B,P»)
 (LAYOUT HIDEROOT
 ("symmetric"«19,18/A,B»)
 (LAYOUT HIDEROOT
 ("A $\in R_{\cup S} \triangleq A \in R \vee A \in S$ "«17,9,3/A,R,S»)))
 (hyp«19/A»)
 (" \vee -E"«15,13,13/A,B,C»)
 (hyp«18/A»)
 (cut«14,13/B,C»)
 ("FROM $S \subseteq T$ INFER $A \in S \rightarrow A \in T$ "«17,9,3/A,S,T»)
 (hyp«16/A»)
 (cut«13,13/B,C»)
 (" \rightarrow -E"«15,13/A,B»)
 (hyp«15/A»)
 (hyp«14/A»)
 (hyp«13/A»)
 (hyp«13/A»)
 (cut«0,4/B,C»)
 (LAYOUT "R_{∪S} = S_{∪R}" ALL

```

("rewrite $\hat{=}$ " $\langle 2, 11, 10, 12/A, xx, B, P \rangle$ )
(LAYOUT HIDEROOT
  ("R $\cup$ S $\hat{=}$ S $\cup$ R" $\langle 9, 3/R, S \rangle$ )
  ("FROM  $cx \in S \vdash cx \in T$  INFER  $S \subseteq T$ " $\langle 8, 3, 10/cx, S, T \rangle$ )
  ("FROM  $A \in S$  INFER  $A \in S \cup T$ " $\langle 8, 3, 9/A, S, T \rangle$ )
  (hyp $\langle 7/A \rangle$ ))
(LAYOUT "S=T $\hat{=}$ S $\subseteq$ T $\wedge$ T $\subseteq$ S" ALL
  ("rewrite $\hat{=}$ " $\langle 4, 5, 6, 5/A, xx, B, P \rangle$ )
  (LAYOUT HIDEROOT
    ("S=T $\hat{=}$ S $\subseteq$ T $\wedge$ T $\subseteq$ S" $\langle 2, 3/S, T \rangle$ )
    (LAYOUT COMPRESS " $\wedge$ -I" ALL
      (" $\wedge$ -I" $\langle 1, 0/A, B \rangle$ )
      (hyp $\langle 1/A \rangle$ )
      (hyp $\langle 0/A \rangle$ )))
  )
  )

```

1.3.3.7. Trees

I didn't make any theorems about trees. That's odd!