COMP0009 Logic — Problem Sheet 4

Part A — Every and some

The story so far. On Sheet 1, Alice sent Bob ten coins over a network where nobody trusts anybody, and every computer on it, Charlie's among them, ran the *script* that locked them. Mallory attacked with scripts that blew up or never stopped, and Charlie's defence was a *typing*, checked before running anything. On Sheet 2 payments came with *certificates*, lists of lines that turned out to be scripts in disguise. On Sheet 3 certificates became *trees*: natural deduction, one bar for each rule, for $\to$, $\wedge$, $\vee$ and $\bot$. Every tree was a program, and Mallory's never-stopping script was a paradox all along.

Now the rulebook grows up. So far it was written in letters, like m: "the coins may move". But the network now carries thousands of payments and files, and a letter cannot say *which*. In the four scenes below the rulebook learns to name things and to say "every" and "some". You will help Charlie read, fill in and build trees for the new rules, and you will catch Mallory in four forgeries, each of which proves something false.

Marks are in square brackets; the whole of Part A is worth 25.

A1. Names inside statements.

The new language has four kinds of symbol.

- *Names* for things: alice, bob, $\mathsf{mallory}$ (users); notes, junk, vault (files); $\mathsf{p7}$ (a payment).
- *Variables* x, y, z, u, v, w: they stand for things nobody has chosen yet.
- *Predicates*, which say something about things: $\mathsf{signed}(\mathsf{p7})$ says "Alice has signed payment $\mathsf{p7}$"; $\mathsf{moves}(\mathsf{p7})$ says "payment $\mathsf{p7}$ may move"; $\mathsf{own}(\mathsf{bob}, \mathsf{notes})$ says "Bob owns notes"; $\mathsf{read}(\mathsf{bob}, \mathsf{notes})$ says "Bob may read notes"; $\mathsf{adm}(\mathsf{alice})$ says "Alice is an administrator".
- Two *quantifiers*: $\forall x\, A$ says "for every x, A", and $\exists x\, A$ says "there is an x such that A".

A statement is built from predicates with $\to$, $\wedge$, $\vee$, $\bot$ and the quantifiers, and $\neg A$ is short for $A \to \bot$, and $\wedge$ and $\vee$ bind more tightly than $\to$, as before. A quantifier reaches only as far as the next statement, so we always write the brackets: $(\forall x\, \mathsf{signed}(x)) \to \mathsf{moves}(x)$ is not $\forall x\, (\mathsf{signed}(x) \to \mathsf{moves}(x))$. The second one says "every payment Alice has signed may move". In $\forall x\, A$ and $\exists x\, A$, the quantifier *binds* every x in A, as λx bound its parameter on Sheet 3. An x that no quantifier binds is *free*. A statement with no free variables is a *sentence*: it says something definite, and only sentences go in the rulebook. Finally, $A(t/x)$ means A with t put in place of every *free* x. If A is $\mathsf{signed}(x) \to \mathsf{moves}(x)$, then $A(\mathsf{p7}/x)$ is $\mathsf{signed}(\mathsf{p7}) \to \mathsf{moves}(\mathsf{p7})$. We only do this when t is a name, or a variable that A has no quantifier on; Part B, B2, shows why.

All of Sheet 3's bars still work. Here is the first new one:

$$\frac{\forall x\, A}{A(t/x)}\ \forall\mathrm{E}$$

"What holds of everything holds of t." Here t is any name, or any variable that A has no quantifier on.

You have used this rule for years. In Haskell, `reverse :: [a] -> [a]` really has the type $\forall a\, ([a] \to [a])$: it works on lists of anything. Each time you call `reverse` on a list of payments, the compiler uses ($\forall$E), with `Payment` for a.

Example. The rulebook says $\forall x\,(\mathsf{signed}(x) \to \mathsf{moves}(x))$, and Alice has signed payment $\mathsf{p7}$. May it move?

$$\dfrac{\dfrac{\forall x\,(\mathsf{signed}(x) \to \mathsf{moves}(x))}{\mathsf{signed}(\mathsf{p7}) \to \mathsf{moves}(\mathsf{p7})}\,\forall\mathrm{E} \qquad \mathsf{signed}(\mathsf{p7})}{\mathsf{moves}(\mathsf{p7})}\,{\to}\mathrm{E}$$

The ($\forall$E) bar puts $\mathsf{p7}$ in place of x, everywhere at once. Then ($\to$E) is modus ponens, as on Sheet 3. Both leaves are open hypotheses: the tree shows

$$\forall x\,(\mathsf{signed}(x) \to \mathsf{moves}(x)),\ \mathsf{signed}(\mathsf{p7}) \vdash_{\mathrm{NJ}} \mathsf{moves}(\mathsf{p7}).$$

Charlie's network decides who may read what with an *access policy* of two lines:

(pol1) $\forall x\,\forall y\,(\mathsf{adm}(x) \to \mathsf{read}(x, y))$ administrators may read anything;
(pol2) $\forall x\,\forall y\,(\mathsf{own}(x, y) \to \mathsf{adm}(x))$ anyone who owns something is an administrator.

(a) Here are four statements:
(1) $\forall y\,(\mathsf{own}(x, y) \to \mathsf{read}(x, y))$
(2) $\exists x\,\mathsf{own}(x, \mathsf{notes})$
(3) $\mathsf{own}(x, \mathsf{notes}) \wedge (\exists x\,\mathsf{read}(x, \mathsf{notes}))$
(4) $\forall x\,\forall y\,(\mathsf{own}(x, y) \to \mathsf{read}(x, y))$
(i) Which variables are free in each? Which of them are sentences? [1]
(ii) Write out $A(\mathsf{bob}/x)$ for each of the four. [1]

Hint: in (3), look at each x separately.

(b) Alice is an administrator: $\mathsf{adm}(\mathsf{alice})$. May she read Bob's notes? Here is Charlie's tree, with two gaps: one statement, and the name of the rule on one bar. Fill them in, and say which t each ($\forall$E) bar used. [1]

$$\dfrac{\dfrac{\dfrac{\forall x\,\forall y\,(\mathsf{adm}(x) \to \mathsf{read}(x, y))}{\boxed{\phantom{\mathsf{adm}(\mathsf{alice}) \to \mathsf{read}(\mathsf{alice}, \mathsf{notes})}}}\,\forall\mathrm{E} \quad \boxed{}}{\mathsf{adm}(\mathsf{alice}) \to \mathsf{read}(\mathsf{alice}, \mathsf{notes})} \qquad \mathsf{adm}(\mathsf{alice})}{\mathsf{read}(\mathsf{alice}, \mathsf{notes})}\,{\to}\mathrm{E}$$

(c) Mallory owns a file of her own, junk: $\mathsf{own}(\mathsf{mallory}, \mathsf{junk})$. Draw a tree for $\mathsf{read}(\mathsf{mallory}, \mathsf{vault})$ whose open hypotheses are (pol1), (pol2) and $\mathsf{own}(\mathsf{mallory}, \mathsf{junk})$. What has Charlie found out about the policy? [2]
Hint: first build a tree for $\mathsf{adm}(\mathsf{mallory})$*, with the pattern of (b) applied to (pol2). Then use (b)'s pattern once more, with your tree in place of the leaf* $\mathsf{adm}(\mathsf{alice})$.

(d) Charlie replaces (pol2) by

(pol3) $\forall x\,\forall y\,(\mathsf{own}(x, y) \to \mathsf{read}(x, y))$ owners may read what they own.

At once Mallory sends him a tree which, she says, shows that owning junk lets her read Bob's notes:

$$\dfrac{\dfrac{\dfrac{\forall x\,\forall y\,(\mathsf{own}(x, y) \to \mathsf{read}(x, y))}{\forall y\,(\mathsf{own}(\mathsf{mallory}, y) \to \mathsf{read}(\mathsf{mallory}, y))}\,\forall\mathrm{E}}{\mathsf{own}(\mathsf{mallory}, \mathsf{junk}) \to \mathsf{read}(\mathsf{mallory}, \mathsf{notes})}\,\forall\mathrm{E} \qquad \mathsf{own}(\mathsf{mallory}, \mathsf{junk})}{\mathsf{read}(\mathsf{mallory}, \mathsf{notes})}\,{\to}\mathrm{E}$$

One bar is wrong. Which, and why? [1]

A2. For everyone.

Charlie wants to make sure that the switch was safe: that (pol3) grants nothing that the old policy did not grant already, that is, that (pol3) follows from (pol1) and (pol2). But (pol3) is a statement about *every* x and y, and ($\forall$E) can only use a $\forall$, not prove one. Here is the rule that proves one:

$$\frac{A(u/x)}{\forall x\, A}\ \forall\text{I}$$

The variable u must not be free in $\forall x\, A$, nor in any hypothesis that is *open* at this bar.

To find the hypotheses *open at a bar*: look at all the leaves above the bar, and cross out the ones discharged by a bar that is also above it. A leaf $[B]^n$ whose bar n is further down is still open here.

The condition says that u is *arbitrary*: nothing the tree still assumes says anything about u. So whatever the tree has shown about u, it could have shown about anything. Proved for an arbitrary u, hence for all.

Example. Charlie's checker has checked every payment Alice signed, and every checked payment may move:

$$\forall x\,(\mathsf{signed}(x) \to \mathsf{checked}(x)) \qquad \forall x\,(\mathsf{checked}(x) \to \mathsf{moves}(x)).$$

So every payment Alice signed may move. To show it, take a payment u about which you know nothing, and suppose Alice signed it:

$$\dfrac{\dfrac{\dfrac{\forall x\,(\mathsf{checked}(x) \to \mathsf{moves}(x))}{\mathsf{checked}(u) \to \mathsf{moves}(u)}\ \forall\text{E} \quad \dfrac{\dfrac{\forall x\,(\mathsf{signed}(x) \to \mathsf{checked}(x))}{\mathsf{signed}(u) \to \mathsf{checked}(u)}\ \forall\text{E} \quad [\mathsf{signed}(u)]^1}{\mathsf{checked}(u)}\ \to\text{E}}{\mathsf{moves}(u)}\ \to\text{E}}{\mathsf{signed}(u) \to \mathsf{moves}(u)}\ \to\text{I},\,1}{\forall x\,(\mathsf{signed}(x) \to \mathsf{moves}(x))}\ \forall\text{I}$$

At the ($\forall$I) bar, the leaves above are the two rules and $[\mathsf{signed}(u)]^1$. Bar 1 is above the ($\forall$I) bar, so it has already discharged $[\mathsf{signed}(u)]^1$. The open hypotheses are the two rules, which are sentences, so u is free in neither, nor in the conclusion. The bar is legal.

(a) (i) Here is Charlie's tree for (pol3) from (pol1) and (pol2), with a run of six gaps down the middle. To save space, the leaves (pol1) and (pol2) stand for the two lines of the policy. Fill in the gaps. [2]

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{(pol1)}}{\forall y\,(\mathsf{adm}(u) \to \mathsf{read}(u,\,y))}\ \forall\text{E}}{\mathsf{adm}(u) \to \mathsf{read}(u,\,v)}\ \forall\text{E} \quad \dfrac{\dfrac{\dfrac{\text{(pol2)}}{\forall y\,(\mathsf{own}(u,\,y) \to \mathsf{adm}(u))}\ \forall\text{E}}{\mathsf{own}(u,\,v) \to \mathsf{adm}(u)}\ \forall\text{E} \quad [\mathsf{own}(u,\,v)]\square}{\square}\ \to\text{E}}{\square}\ \to\text{E}}{\square}\ \to\text{I},\square}{\square}\ \forall\text{I}}{\forall x\,\forall y\,(\mathsf{own}(x,\,y) \to \mathsf{read}(x,\,y))}\ \forall\text{I}$$

(ii) At each of the two ($\forall$I) bars, which hypotheses are open? Check that the bar is legal. [1]

Hint: fill the gaps from the top down. The leaf $[\mathsf{own}(u,\,v)]$ *is discharged by the bar whose conclusion begins with* $\mathsf{own}(u,\,v) \to$.

(b) The policy is now (pol1) and (pol3), and Alice is an administrator. Mallory sends Charlie this tree, whose open hypotheses are (pol1) and $\mathsf{adm}(\mathsf{alice})$. It concludes that Mallory may read the vault, because Alice is an administrator.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x\,\forall y\,(\mathsf{adm}(x) \to \mathsf{read}(x,\,y))}{\forall y\,(\mathsf{adm}(u) \to \mathsf{read}(u,\,y))}\,\forall\mathrm{E}}{\mathsf{adm}(u) \to \mathsf{read}(u,\,\mathsf{vault})}\,\forall\mathrm{E} \qquad [\mathsf{adm}(u)]^1}{\mathsf{read}(u,\,\mathsf{vault})}\,{\to}\mathrm{E}}{\forall x\,\mathsf{read}(x,\,\mathsf{vault})}\,\forall\mathrm{I}}{\mathsf{adm}(u) \to \forall x\,\mathsf{read}(x,\,\mathsf{vault})}\,{\to}\mathrm{I},\,1}{\forall y\,(\mathsf{adm}(y) \to \forall x\,\mathsf{read}(x,\,\mathsf{vault}))}\,\forall\mathrm{I}}{\mathsf{adm}(\mathsf{alice}) \to \forall x\,\mathsf{read}(x,\,\mathsf{vault})}\,\forall\mathrm{E} \qquad \mathsf{adm}(\mathsf{alice})}{\forall x\,\mathsf{read}(x,\,\mathsf{vault})}\,{\to}\mathrm{E}}{\mathsf{read}(\mathsf{mallory},\,\mathsf{vault})}\,\forall\mathrm{E}$$

(i) One bar breaks the side condition of its rule. Which bar, and why? [1]

(ii) The other ($\forall$I) bar, the one that concludes a statement beginning $\forall y$, is legal. Why? Say in plain English what the bad bar claimed, and why it does not follow. [1]

(c) The same side condition is in your compiler. Here is a Haskell function, and what GHC says about it:

```
f :: a -> b
f x = x
```

GHC:
```
Couldn't match expected type ‘b’ with actual type ‘a’
    ‘a’ is a rigid type variable bound by
        the type signature for: f :: forall a b. a -> b
```

The signature promises a function of type $\forall a\,\forall b\,(a \to b)$. While GHC checks the body, $x : a$ is in its *context*, the list of names it has been told about (as on Sheet 3), just as $[\mathsf{adm}(u)]^1$ was open at Mallory's bad bar. Here are two bars that would turn $x : a$ into $x : b$:

$$\dfrac{\dfrac{x : a}{x : \forall a\,a}\,\forall\mathrm{I}}{x : b}\,\forall\mathrm{E}$$

Which bar of Mallory's tree is each of them like? Which of the two is illegal, and so forbidden by what GHC calls "rigid"? [1]

A3. Someone.

Alice's coins are kept in a vault, and the rulebook says that Bob holds a key that opens it:

$$\text{(key)}\quad \exists x\,(\mathsf{holds}(\mathsf{bob},\,x) \wedge \mathsf{opens}(x,\,\mathsf{vault})).$$

It does not say which key; Bob keeps that to himself. Here are the two rules for $\exists$:

$$\dfrac{A(t/x)}{\exists x\,A}\,\exists\mathrm{I}$$

"t has the property, so something does." Here t is any name, or any variable that A has no quantifier on.

$$\dfrac{\exists x\,A \qquad \begin{array}{c}[A(u/x)]^n \\ \vdots \\ C\end{array}}{C}\,\exists\mathrm{E},\,n$$

The variable u must not be free in C, nor in $\exists x\,A$, nor in any hypothesis open at this bar other than the leaves $[A(u/x)]^n$.

To *prove* "there is", you must name one: from $\mathsf{opens}(\mathsf{k1}, \mathsf{vault})$, ($\exists$I) gives $\exists x\, \mathsf{opens}(x, \mathsf{vault})$. To *use* "there is a thing with A", call the thing by a new variable, u, which appears nowhere else; suppose $A(u/x)$, and argue to C. The bar discharges the leaves $[A(u/x)]^n$, like ($\vee$E) on Sheet 3. Then u may not leave the bar: you made the variable up, so C may not mention it. As on Sheet 3, every tree is a program: an ($\exists$I) bar packs the witness with its evidence, like a pair, and an ($\exists$E) bar unpacks it, like a **case**.

Example. From the rulebook's line about Bob, Charlie shows that some key opens the vault. Call Bob's key u:

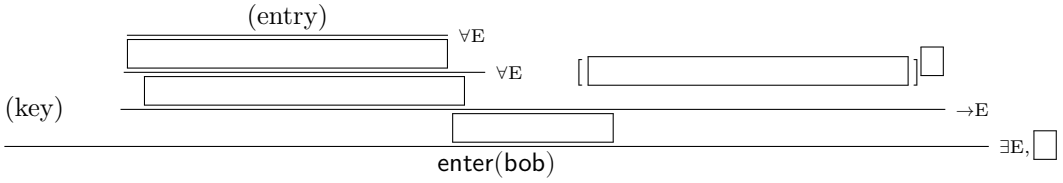
$$\dfrac{\exists x\,(\mathsf{holds}(\mathsf{bob}, x) \wedge \mathsf{opens}(x, \mathsf{vault})) \qquad \dfrac{\dfrac{[\mathsf{holds}(\mathsf{bob}, u) \wedge \mathsf{opens}(u, \mathsf{vault})]^1}{\mathsf{opens}(u, \mathsf{vault})}\,\wedge\mathrm{E}_2}{\exists x\, \mathsf{opens}(x, \mathsf{vault})}\,\exists\mathrm{I}}{\exists x\, \mathsf{opens}(x, \mathsf{vault})}\,\exists\mathrm{E},1$$

At the ($\exists$E) bar: u is not in the conclusion $\exists x\, \mathsf{opens}(x, \mathsf{vault})$ (the x there is bound), nor in the rulebook's line, and no other hypothesis is open. The bar is legal.

(a) The rulebook also says that anyone who holds a key that opens the vault may enter it:

(entry) $\quad \forall x\, \forall y\, (\mathsf{holds}(x, y) \wedge \mathsf{opens}(y, \mathsf{vault}) \to \mathsf{enter}(x))$.

Here is the shape of a tree for $\mathsf{enter}(\mathsf{bob})$ whose open hypotheses are (entry) and (key): only the bars, their rule names, those two leaves and the bottom statement are given. Fill in the six boxes, and check the three conditions at the ($\exists$E) bar. [2]



Hint: the right-hand branch takes the rule apart with two ($\forall$E) bars, at bob *and at* u.

(b) An auditor asks Charlie: "Under the *old* policy, (pol1) and (pol2), and given that Mallory owns junk, could anyone read the vault?" That is $\exists x\, \mathsf{read}(x, \mathsf{vault})$. Add one bar to your tree of A1(c) to prove it. Which name must the new bar use? What does the tree tell the auditor that a bare "yes" would not? [1]

(c) Mallory reads a line of the rulebook, $\exists x\, \mathsf{opens}(x, \mathsf{vault})$, and sends Charlie this tree. It concludes that her own key, mkey, opens the vault.

$$\dfrac{\dfrac{\dfrac{\exists x\, \mathsf{opens}(x, \mathsf{vault}) \qquad [\mathsf{opens}(u, \mathsf{vault})]^1}{\mathsf{opens}(u, \mathsf{vault})}\,\exists\mathrm{E},1}{\forall x\, \mathsf{opens}(x, \mathsf{vault})}\,\forall\mathrm{I}}{\mathsf{opens}(\mathsf{mkey}, \mathsf{vault})}\,\forall\mathrm{E}$$

(i) One bar breaks a side condition. Which bar, and which of its three conditions? [1]

(ii) The ($\forall$I) bar is the one that concludes that *every* key opens the vault. Why is it not to blame? [1]

(d) Every key Bob holds opens the vault, $\forall x\, (\mathsf{holds}(\mathsf{bob}, x) \to \mathsf{opens}(x, \mathsf{vault}))$, and Bob holds a key, $\exists x\, \mathsf{holds}(\mathsf{bob}, x)$. Draw a tree for $\exists x\, \mathsf{opens}(x, \mathsf{vault})$ whose open hypotheses are these two. [1]

A4. On your own.

On the network, users vouch for each other's keys: $\mathsf{vouch}(x,\, y)$ says that x vouches for y. From here on you draw every tree yourself. A method that usually works: start from the bottom, and look at the statement you want and the hypotheses you have.

- To prove $A \to B$: suppose $[A]^n$ and aim for B; finish with ($\to$I).
- To prove $\forall x\, A$: aim for $A(u/x)$, for a variable u used nowhere else; finish with ($\forall$I).
- To prove $\exists x\, A$: find the right t, aim for $A(t/x)$, and finish with ($\exists$I).
- To use $\forall x\, A$: take it apart with ($\forall$E), at whatever t you need.
- To use $\exists x\, A$: open it with ($\exists$E) *early*, near the bottom of the tree, with a new variable u; then work above it with $[A(u/x)]^n$ as a leaf.

(a) An *authority* is someone who vouches for everybody: $\exists x\, \forall y\, \mathsf{vouch}(x,\, y)$ says there is one. Draw a tree, with no open hypotheses, for

$$(\exists x\, \forall y\, \mathsf{vouch}(x,\, y)) \to \forall y\, \exists x\, \mathsf{vouch}(x,\, y)\ :$$

if there is an authority, then everybody is vouched for by somebody. [2]

Hint: suppose $[\exists x\, \forall y\, \mathsf{vouch}(x,\, y)]^2$. *Aim for* $\exists x\, \mathsf{vouch}(x,\, v)$, *opening the hypothesis with (*$\exists$*E) and the leaf* $[\forall y\, \mathsf{vouch}(u,\, y)]^1$*; then generalise over* v.

(b) Mallory claims the converse: if everybody is vouched for by somebody, then there is an authority. Here is her tree:

$$\dfrac{\dfrac{\dfrac{[\forall y\, \exists x\, \mathsf{vouch}(x,\, y)]^2}{\exists x\, \mathsf{vouch}(x,\, v)}\ \forall\mathrm{E} \qquad \dfrac{\dfrac{[\mathsf{vouch}(u,\, v)]^1}{\forall y\, \mathsf{vouch}(u,\, y)}\ \forall\mathrm{I}}{\exists x\, \forall y\, \mathsf{vouch}(x,\, y)}\ \exists\mathrm{I}}{\exists x\, \forall y\, \mathsf{vouch}(x,\, y)}\ \exists\mathrm{E},\,1}{(\forall y\, \exists x\, \mathsf{vouch}(x,\, y)) \to \exists x\, \forall y\, \mathsf{vouch}(x,\, y)}\ \to\mathrm{I},\,2$$

(i) Which bar breaks a side condition, and how? [1]

(ii) On a small network there are just two users: Alice vouches for Bob, Bob vouches for Alice, and nobody vouches for themselves. Is Mallory's statement true on this network? Take on trust that a tree with no open hypotheses never proves a statement that is false on some network. Can anyone repair Mallory's tree? [1]

(c) Alice wants a rule: "if anyone reads the vault, the alarm sounds". Write alarm for "the alarm sounds", a statement with no variables. Charlie and Bob write it in two ways:

$$\text{Charlie:}\quad \forall x\, (\mathsf{read}(x,\, \mathsf{vault}) \to \mathsf{alarm})$$
$$\text{Bob:}\quad (\exists x\, \mathsf{read}(x,\, \mathsf{vault})) \to \mathsf{alarm}$$

(i) Draw a tree for Bob's version whose only open hypothesis is Charlie's. [1]

Hint: the ($\exists$*E) bar's conclusion is* alarm*, which has no* u *in it.*

(ii) Draw a tree for Charlie's version whose only open hypothesis is Bob's. [1]

Hint: suppose $\mathsf{read}(u,\, \mathsf{vault})$ *and build* $\exists x\, \mathsf{read}(x,\, \mathsf{vault})$ *from it.*

(iii) Mallory offers a third version, $(\forall x\, \mathsf{read}(x,\, \mathsf{vault})) \to \mathsf{alarm}$. When does her alarm sound? One of the two directions between Charlie's and Mallory's versions has a tree: which? [1]

Charlie's questions are real. Amazon Web Services checks access policies with a tool called Zelkova, which translates a policy into logic and hands questions about it to automated solvers (Backes and others, FMCAD 2018). Amazon S3, for example, asks it: "Does this S3 bucket policy grant public access?", which is the auditor's question of A3(b). AWS's policy language Cedar (OOPSLA 2024) was designed to be analysed this way, and some of its key properties, such as "forbid overrides permit", are proved in the proof assistant Lean. In Lean itself, a proof of $\exists x\, A$ is built as a pair $\langle$`w, h`$\rangle$ of a witness and its evidence, and used with `Exists.elim`, which asks for a proof that works "for an arbitrary value `w`": ($\exists$I) and ($\exists$E). And the side condition of A2 is in the Haskell compiler, whose "rigid type variable" is a type variable that the context has fixed, and that the checker may therefore neither generalise over nor replace.

Part B — Practice

Practice, for when you want more of the same; nothing here is marked. The drills go from reading statements to building trees and checking them. The last two, B13 and B14, go beyond this week: try them only if you are curious. In the drills, P, Q and R are predicates, C is a statement with no variables, c and d are names, and x, y, z, u, v, w are variables. The rules you need:

- Statements: predicates applied to names and variables, $s = t$, $\bot$, $A \to B$, $A \wedge B$, $A \vee B$, $\forall x\, A$ and $\exists x\, A$. $\neg A$ is short for $A \to \bot$. A quantifier reaches only as far as the next statement, so brackets are always written: $(\forall x\, P(x)) \to Q(x)$ is not $\forall x\, (P(x) \to Q(x))$.
- In $\forall x\, A$ and $\exists x\, A$ the quantifier binds every x in A; an x that no quantifier binds is *free*; a statement with no free variables is a *sentence*. $A(t/x)$ is A with t put in place of every free x; we do this only when t is a name, or a variable that A has no quantifier on.
- Trees: a leaf is a hypothesis; $[A]^n$ is a leaf discharged by the bar numbered n below it. The hypotheses *open at a bar* are the leaves above it, except those discharged by a bar that is also above it. A tree with open hypotheses Γ and A at the bottom shows $\Gamma \vdash_{\mathrm{NJ}} A$.

Sheet 3's bars.

$$\frac{A \to B \qquad A}{B}\ {\to}\mathrm{E} \qquad \frac{\begin{array}{c}[A]^n\\ \vdots\\ B\end{array}}{A \to B}\ {\to}\mathrm{I}, n \qquad \frac{A \qquad B}{A \wedge B}\ {\wedge}\mathrm{I} \qquad \frac{A \wedge B}{A}\ {\wedge}\mathrm{E}_1 \qquad \frac{A \wedge B}{B}\ {\wedge}\mathrm{E}_2$$

$$\frac{A}{A \vee B}\ {\vee}\mathrm{I}_1 \qquad \frac{B}{A \vee B}\ {\vee}\mathrm{I}_2 \qquad \frac{\begin{array}{ccc} & [A]^i & [B]^j\\ A \vee B & \vdots & \vdots\\ & C & C\end{array}}{C}\ {\vee}\mathrm{E}, i, j \qquad \frac{\bot}{C}\ \bot\mathrm{E}$$

A ($\to$I) bar may also discharge no leaves at all.

The new bars.

$$\frac{\forall x\, A}{A(t/x)}\ \forall\mathrm{E} \qquad \frac{A(t/x)}{\exists x\, A}\ \exists\mathrm{I}$$

For both: t is a name, or a variable that A has no quantifier on.

$$\frac{A(u/x)}{\forall x\, A}\ \forall\mathrm{I}$$

u is a variable that is not free in $\forall x\, A$, nor in any hypothesis open at the bar.

$$\frac{\exists x\, A \qquad \begin{array}{c}[A(u/x)]^n\\ \vdots\\ C\end{array}}{C}\ \exists\mathrm{E}, n$$

u is a variable that is not free in C, nor in $\exists x\, A$, nor in any hypothesis open at the bar other than the leaves $[A(u/x)]^n$.

$$\frac{}{t = t}\ =\mathrm{I} \qquad \frac{s = t \qquad A(s/x)}{A(t/x)}\ =\mathrm{E}$$

(=I) has no premises. (=E) puts equals for equals, in any statement A.

B1. Reading statements. For (i)–(iv), list the free variables and say which are sentences. For (v)–(viii), read $R(x,\, y)$ as "x trusts y", and say each in plain English.

(i) $P(x) \wedge (\forall x\, Q(x))$
(ii) $\forall x\, (P(x) \to \exists y\, R(x,\, y))$
(iii) $(\exists y\, R(x,\, y)) \to \forall x\, R(x,\, y)$
(iv) $\forall x\, R(x,\, y)$
(v) $\forall x\, \exists y\, R(x,\, y)$
(vi) $\exists y\, \forall x\, R(x,\, y)$
(vii) $\exists x\, \forall y\, R(x,\, y)$
(viii) $\forall x\, \neg R(x,\, x)$

B2. Substitution. Write out each, or say why it is not allowed.

(i) $A(c/x)$, where A is $P(x) \wedge (\forall x\, Q(x))$.
(ii) $A(c/x)$ and $A(z/x)$, where A is $\exists y\, R(x,\, y)$.
(iii) $A(y/x)$, where A is $\exists y\, R(x,\, y)$. If you put y in blindly, what does the result say, and what should it say? How do you put it right?
(iv) In Haskell, `map :: (a -> b) -> [a] -> [b]` has the type $\forall a\, \forall b\, ((a \to b) \to [a] \to [b])$. The call `map show [1, 2]` uses it at `show :: Int -> String`. Which ($\forall$E) bars does the compiler use, with which t, and what type do they give?

B3. Reading trees. For each tree, say which hypotheses are open, write what it shows as "$\Gamma \vdash_{\mathrm{NJ}} A$", and check the side condition at each ($\forall$I) and ($\exists$E) bar.

(i)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x\, (P(x) \wedge Q(x))}{P(u) \wedge Q(u)}\,\forall\mathrm{E}}{P(u)}\,\wedge\mathrm{E}_1}{\forall x\, P(x)}\,\forall\mathrm{I} \qquad \dfrac{\dfrac{\dfrac{\forall x\, (P(x) \wedge Q(x))}{P(u) \wedge Q(u)}\,\forall\mathrm{E}}{Q(u)}\,\wedge\mathrm{E}_2}{\forall x\, Q(x)}\,\forall\mathrm{I}}{(\forall x\, P(x)) \wedge (\forall x\, Q(x))}\,\wedge\mathrm{I}$$

(ii)

$$\dfrac{\exists x\, P(x) \qquad \dfrac{\dfrac{[P(u)]^1}{P(u) \vee Q(u)}\,\vee\mathrm{I}_1}{\exists x\, (P(x) \vee Q(x))}\,\exists\mathrm{I}}{\exists x\, (P(x) \vee Q(x))}\,\exists\mathrm{E},1$$

B4. "$\exists$ over $\vee$". Fill in the eight gaps of this tree, which shows $\exists x\, (P(x) \vee Q(x)) \vdash_{\mathrm{NJ}} (\exists x\, P(x)) \vee (\exists x\, Q(x))$:

$$\dfrac{\exists x\, (P(x) \vee Q(x)) \qquad \dfrac{[\qquad\qquad]^{\square} \qquad \dfrac{\dfrac{[P(u)]^2}{\qquad\qquad}\,\exists\mathrm{I}}{(\exists x\, P(x)) \vee (\exists x\, Q(x))}\,\vee\mathrm{I}_1 \qquad \dfrac{\dfrac{[\qquad\qquad]^3}{\exists x\, Q(x)}\,\square}{(\exists x\, P(x)) \vee (\exists x\, Q(x))}\,\vee\mathrm{I}_2}{(\exists x\, P(x)) \vee (\exists x\, Q(x))}\,\vee\mathrm{E},\square,\square}{(\exists x\, P(x)) \vee (\exists x\, Q(x))}\,\exists\mathrm{E},\square$$

B5. Draw a tree with no open hypotheses for each.

(i) $(\forall x\, P(x)) \wedge (\forall x\, Q(x)) \to \forall x\, (P(x) \wedge Q(x))$
(ii) $(\exists x\, (P(x) \wedge Q(x))) \to (\exists x\, P(x)) \wedge (\exists x\, Q(x))$

(iii) The converse of (ii) has no tree. Find two things and a choice of P and Q for which it is false. Where does an attempt at a tree get stuck?

B6. Swapping quantifiers. Draw a tree with no open hypotheses for each.

(i) $(\forall x\, \forall y\, R(x,\, y)) \to \forall y\, \forall x\, R(x,\, y)$ (ii) $(\exists x\, \exists y\, R(x,\, y)) \to \exists y\, \exists x\, R(x,\, y)$

Hint: in (ii), open both ∃s with (∃E) bars near the bottom, with two different variables.

B7. The old access policy again, and a fact:

(pol1) $\forall x\, \forall y\, (\mathsf{adm}(x) \to \mathsf{read}(x,\, y))$
(pol2) $\forall x\, \forall y\, (\mathsf{own}(x,\, y) \to \mathsf{adm}(x))$
(fact) $\mathsf{own}(\mathsf{bob},\, \mathsf{notes})$ Bob owns his notes.

(i) Draw a tree for $\mathsf{read}(\mathsf{bob},\, \mathsf{notes})$ whose open hypotheses are (pol1), (pol2) and $\mathsf{own}(\mathsf{bob},\, \mathsf{notes})$.
(ii) Add one bar to get $\exists x\, \mathsf{read}(x,\, \mathsf{notes})$.
(iii) Write as a statement: "nobody may read the log, except administrators", using a name log. Why is $\forall x\, (\mathsf{adm}(x) \to \mathsf{read}(x,\, \mathsf{log}))$ not it?

B8. "Nobody may read the vault" can be written $\neg\exists x\, P(x)$ or $\forall x\, \neg P(x)$, with $P(x)$ for $\mathsf{read}(x,\, \mathsf{vault})$. Show that the two say the same thing: draw a tree with no open hypotheses for each of

(i) $\neg\exists x\, P(x) \to \forall x\, \neg P(x)$ (ii) $(\forall x\, \neg P(x)) \to \neg\exists x\, P(x)$

Hint: $\neg A$ is $A \to \bot$: to prove it, suppose A and aim for $\bot$.

B9. Equality. Draw a tree with no open hypotheses for each:

(i) $\forall x\, \forall y\, (x = y \to y = x)$ (ii) $\forall x\, \forall y\, \forall z\, (x = y \wedge y = z \to x = z)$

Why are these two not rules of their own?

Hint: for (i), suppose $u = v$. Use (=E) with A the statement $z = u$: then $A(u/z)$ is $u = u$, which (=I) gives, and $A(v/z)$ is $v = u$.

B10. Draw a tree with no open hypotheses for each:

(i) $\exists x\, (x = x)$ (ii) $(\exists x\, P(x)) \to \neg\forall x\, \neg P(x)$

What does (i) say about the world, and why might that be a surprise?

B11. Checking trees. Each tree below has exactly one mistake. Find the bar that is wrong, and say which condition it breaks.

(i) A tree that claims $\forall x\, \neg\forall y\, (y = x) \vdash_{\mathrm{NJ}} \bot$, though its hypothesis says only that nothing is the only thing there is:

$$\dfrac{\dfrac{\forall x\, \neg\forall y\, (y = x)}{\neg\forall y\, (y = y)}\ \forall\mathrm{E} \qquad \dfrac{\dfrac{}{y = y}\ =\mathrm{I}}{\forall y\, (y = y)}\ \forall\mathrm{I}}{\bot}\ \to\mathrm{E}$$

(ii) A tree that claims $\exists x\, P(x),\ Q(u) \vdash_{\mathrm{NJ}} \exists x\, (P(x) \wedge Q(x))$:

$$\dfrac{\exists x\, P(x) \qquad \dfrac{\dfrac{[P(u)]^1 \qquad Q(u)}{P(u) \wedge Q(u)}\,\wedge\mathrm{I}}{\exists x\, (P(x) \wedge Q(x))}\,\exists\mathrm{I}}{\exists x\, (P(x) \wedge Q(x))}\,\exists\mathrm{E},\,1$$

(iii) A tree that claims $\vdash_{\mathrm{NJ}} \forall y\, \forall x\, (x = y)$: everything equals everything.

$$\dfrac{\dfrac{\dfrac{}{u = u}\,=\mathrm{I}}{\forall x\, (x = u)}\,\forall\mathrm{I}}{\forall y\, \forall x\, (x = y)}\,\forall\mathrm{I}$$

(iv) A tree that claims $\exists x\, R(x,\, u) \vdash_{\mathrm{NJ}} \exists x\, R(x,\, x)$: if somebody trusts u, then somebody trusts themselves.

$$\dfrac{\exists x\, R(x,\, u) \qquad \dfrac{[R(u,\, u)]^1}{\exists x\, R(x,\, x)}\,\exists\mathrm{I}}{\exists x\, R(x,\, x)}\,\exists\mathrm{E},\,1$$

B12. A quantifier inside an implication. Draw a tree with no open hypotheses for each, and say each in plain English, with $P(x)$ for "x has signed" and $Q(x)$ for "x may vote".

(i) $(\forall x\, (P(x) \to Q(x))) \to (\forall x\, P(x)) \to \forall x\, Q(x)$
(ii) $(\forall x\, (P(x) \to Q(x))) \to (\exists x\, P(x)) \to \exists x\, Q(x)$
(iii) $(\exists x\, (P(x) \to C)) \to (\forall x\, P(x)) \to C$

B13. *(For the curious.)* Suppose ($\forall$I) had no side condition.

(i) From the hypothesis $u = c$, derive $\forall x\, (x = c)$ in one bar. Then derive, with no open hypotheses, first

$$u = c \to \forall x\, (x = c) \qquad \text{and then} \qquad \forall x\, (x = c).$$

(ii) Conclude $\bot$ from the hypothesis $\neg\forall x\, (x = c)$, which says only that c is not the only thing there is.
(iii) Which bar does the real rule forbid? What would the Haskell compiler call u?

B14. *(For the curious.)* B10(ii) showed $(\exists x\, P(x)) \to \neg\forall x\, \neg P(x)$. Try to draw a tree with no open hypotheses for the converse, $\neg\forall x\, \neg P(x) \to \exists x\, P(x)$, and say where every attempt gets stuck. Compare A3(b): what would a tree for it have to do that the hypothesis cannot?