Whither Inference Rules?

We assume the following Propositional Calculus rules:

(weakening)	$A \wedge B$	$\Rightarrow$	A
(absorption)	$A \wedge (A \Rightarrow B)$	$\equiv$	$A \wedge B$
(trading)	$A \wedge B \Rightarrow C$	$\equiv$	$A \Rightarrow (B \Rightarrow C)$

(legenda: $\wedge$ binds stronger than $\Rightarrow$, and $\Rightarrow$ binds stronger than $\equiv$.)

With these rules the following lemma can be proved:

<u>lemma</u> $(A \Rightarrow B) \Rightarrow (A \Rightarrow C) \quad \equiv \quad A \Rightarrow (B \Rightarrow C)$

<u>proof</u>:

$$
\begin{array}{ll}
 & (A \Rightarrow B) \Rightarrow (A \Rightarrow C) \\
\equiv & \quad \{ \text{ trading } \} \\
 & (A \Rightarrow B) \wedge A \Rightarrow C \\
\equiv & \quad \{ \text{ absorption } (\wedge \text{ is symmetric}) \} \\
 & A \wedge B \Rightarrow C \\
\equiv & \quad \{ \text{ trading } \} \\
 & A \Rightarrow (B \Rightarrow C)
\end{array}
$$

□

The relevance of the lemma is twofold. First, it offers a possibility for simplification: when applied from left to right the number of occurrences of A decreases by one. Second, it factors out the occurrences of A; this is nice in those cases where $B \Rightarrow C$ can be established in isolation, i.e., independently of A. The effect of simplification becomes more convincingly visible in more complicated examples, such as:

$$
\begin{array}{ll}
 & (A \Rightarrow B) \wedge (A \Rightarrow C) \;\Rightarrow\; (A \Rightarrow D) \\
\equiv & \quad \{ \text{ trading } \} \\
 & (A \Rightarrow B) \;\Rightarrow\; ((A \Rightarrow C) \Rightarrow (A \Rightarrow D)) \\
\equiv & \quad \{ \text{ lemma } \} \\
 & (A \Rightarrow B) \;\Rightarrow\; (A \Rightarrow (C \Rightarrow D)) \\
\equiv & \quad \{ \text{ lemma } \} \\
 & A \;\Rightarrow\; (B \Rightarrow (C \Rightarrow D)) \\
\equiv & \quad \{ \text{ trading } \} \\
 & A \;\Rightarrow\; (B \wedge C \Rightarrow D) \quad .
\end{array}
$$

In this case, three occurrences of A have been reduced to one.

* * *

We apply the above to an (arbitrarily chosen) collection of type-inference rules for the λ-calculus. Our concern here is the shape of these rules, not the question whether they yield a meaningful typing system. In what follows v ranges over the variables of the λ-calculus, E and F denote expressions of the λ-calculus, T and U denote types, and C denotes a so-called environment —explanation follows— . The type-inference rules are:

(0) $$\frac{(v : T) \in C}{C \vdash (v : T)}$$

(1) $$\frac{C \vdash (E : T) \quad , \quad T \leq U}{C \vdash (E : U)}$$

(2) $$\frac{\{(v : T)\} \cup C \quad \vdash \quad (E : U)}{C \vdash (\lambda v . E \ : \ T \to U)}$$

(3) $$\frac{C \vdash (E : T \to U) \quad , \quad C \vdash (F : T)}{C \vdash (EF \ : U)}$$

explanation: For types T, U we have that $T \to U$ is a type as well; $\leq$ is a binary relation on types, so $T \leq U$ is a proposition. For expression E and type T we have that $(E : T)$ is a proposition. An environment C is a set of so-called type assignments, which are propositions of the shape $(v : T)$.

□

The symbol $\in$ in (0) can be done away with by rewriting (0) as follows:

(0') $$\frac{}{\{(v : T)\} \cup C \quad \vdash \ (v : T)}$$

Both the symbols ——————— and $\vdash$ represent a form of logical implication. Both the symbols $,$ and $\cup$ represent a form of logical conjunction. An environment represents a proposition, namely the conjunction of all propositions contained in that environment. With $\Rightarrow$ for (both forms of) implication, with $\wedge$ for (both forms of) conjunction, and with C now for any proposition, the above inference rules can be reformulated as:

(0a) $\quad true \;\Rightarrow\; ((v:T) \wedge C \;\Rightarrow\; (v:T))$

(1a) $\quad (C \Rightarrow (E:T)) \;\wedge\; T \leq U \;\;\Rightarrow\;\; (C \Rightarrow (E:U))$

(2a) $\quad ((v:T) \wedge C \;\Rightarrow\; (E:U)) \;\;\Rightarrow$
$\qquad (C \;\Rightarrow\; (\lambda v.E : T \to U))$

(3a) $\quad (C \Rightarrow (E : T \to U)) \;\wedge\; (C \Rightarrow (F:T)) \;\;\Rightarrow$
$\qquad (C \Rightarrow (EF : U))$

Because "true $\Rightarrow$" may be omitted we conclude that rule (0a) is void: it is just an instance of the weakening rule $A \wedge B \Rightarrow A$. In the original collection of rules, rule (0) (or (0')) nevertheless is indispensable because it is the only rule by means of which formulae of the form $C \vdash P$ can be introduced.

The above rules contain many "uninterpreted" occurrences of C. Here our lemma enters the picture. By application of the lemma, and of the trading rule, further simplification is possible:

(1b) $\quad C \;\Rightarrow\; ((E:T) \wedge T \leq U \;\Rightarrow\; (E:U))$

(2b) $\quad C \;\Rightarrow\; (((v:T) \Rightarrow (E:U)) \;\Rightarrow\; (\lambda v.E : T \to U))$

(3b) $C \Rightarrow ((E : T \to U) \wedge (F : T) \Rightarrow (EF : U))$

Finally, we even go one step further. The above rules —as is always the case with inference rules— are, albeit implicitly, universally quantified over the variables occurring in them. In particular, the above rules hold for all C. Therefore, we may omit the antecedents C altogether*):

(1c) $(E : T) \wedge T \leq U \Rightarrow (E : U)$

(2c) $((v : T) \Rightarrow (E : U)) \Rightarrow (\lambda v . E : T \to U)$

(3c) $(E : T \to U) \wedge (F : T) \Rightarrow (EF : U)$

Not only have the rules been simplified, in this form they are also better geared for use in proofs by calculation. For example, if so desired (1c) can be transformed, by trading, into the nicer:

(1d) $T \leq U \Rightarrow ((E : T) \Rightarrow (E : U))$

Dropping the environments C not only simplifies the rules, it also enhances proof modularisation —separation of concerns—. Rule (3c), for example, allows us to conclude (EF : U) from (E : T→U) and (F : T) independently of how the latter are or will be proved; that is, for the validity of (3c) it is absolutely irrelevant whether (E : T→U) and (F : T) are proved or are established

*) without any loss, that is.

by magic, so to speak.

It may very well be that I have overlooked the essence and that I have thrown away the baby with the bathwater. Yet, if there is a reason to prefer formulae (0) through (3) over the much simpler (1c) through (3c), it had better be a very good one, for the price for undue complexity is high. If such a reason exists I will be very glad to be informed about it.

Eindhoven, 9 June 1992

Rob R. Hoogerwoord
dept. of mathematics and computing science
Eindhoven University of Technology
postbus 513
5600 MB Eindhoven