Theoretical Computer Science 28 (1984) 235–236
North-Holland

NOTE

COPPO–DEZANI TYPES DO NOT CORRESPOND TO PROPOSITIONAL LOGIC

J. Roger HINDLEY

Department of Mathematics, University College of Swansea, Singleton Park, Swansea SA2 8PP, United Kingdom

Communicated by M. Nivat
Received June 1983

Abstract. This note gives a simple counterexample to back up Pottinger's explanation of the difference between propositional logic, and the type theories of Pottinger (1980) and Coppo–Dezani (1978). The example depends on a theorem of Ben-Yelles (1979).

Several people in computer science have asked me whether the type-schemes for λ-calculus that Coppo and Dezani introduced in [3] correspond to provable formulae in some propositional logic. The answer is "no", and was given by Pottinger in [4] for his similar system. The present note backs up Pottinger's answer with a specific counterexample based on a theorem of Ben-Yelles [2].

I shall use here the notation of [1]. Coppo–Dezani type schemes are given in [1, Definition 2.1] and the rules for assigning them to λ-terms are given in [1, Definition 2.5]. The set of all type schemes which are assigned to closed λ-terms will be called S:

$$S=\{\sigma\colon (\exists \text{ closed } M)(\vdash \sigma M)\}.$$

In the Curry system [1, Section 1], the only type-forming connective is '$\rightarrow$', and S coincides with the set of all provable formulae of intuitionist implicational logic.

Coppo–Dezani type schemes use '$\cap$' (intersection) as well as '$\rightarrow$'. The introduction and elimination rules for '$\cap$' are very like those for '$\wedge$' (conjunction) in logic, and it is tempting to think that S becomes the set of provable formulae of some system of logic when '$\cap$' is interpreted as '$\wedge$'.

This is not the case. The reason was stated by Pottinger in [4, p. 561]: In the Coppo–Dezani $\cap$-introduction rule,

$$\sigma M,\ \tau M\ \vdash\ (\sigma\cap\tau)M,$$

M is the same in the conclusion as in both premises (and indeed must be so, if '$\cap$'

0304-3975/84/$3.00

is to represent intersection). Thus M does not grow as the deduction grows, and so λ-terms do not correspond to deductions, as they do in the Curry system.

The following example backs up Pottinger's remark.

Example. *Let a, b, c be distinct type-variables, and define*

$$\sigma = ((a \to a) \cap ((a \to .b \to c) \to (a \to b. \to .a \to c))).$$

The corresponding propositional formula (with '$\wedge$' for '$\cap$') is provable in intuitionistic logic, and in most other logics too. But $\sigma \notin S$.

Proof. Ben-Yelles [2, Theorem 4.49] says that, for the Curry system, if

$$\vdash (a \to a)M, \tag{1}$$

then $M =_\beta \lambda x.x$, and if

$$\vdash ((a \to .b \to c) \to (a \to b. \to .a \to c))M, \tag{2}$$

then $M =_\beta \lambda xyz.xz(yz)$. Now let σM be provable in the Coppo–Dezani system. Then by the $\cap$-elimination rule, (1) and (2) hold in that system. By the conservative-extension theorem [1, Corollary 4.10], they hold for the Curry system too. Hence M has two normal forms, which is impossible.

References

[1] H.P. Barendregt, M. Coppo and M. Dezani-Ciancaglini, A filter lambda model and the completeness of type assignment, *J. Symbolic Logic* (1983) to appear.

[2] C.-B. Ben-Yelles, Type-assignment in the lambda calculus, Ph.D. Thesis, Institut de Mathematiques, Université des Sciences, Alger, Algérie, 1979.

[3] M. Coppo and M. Dezani-Ciancaglini, A new type-assignment for λ-terms, *Archiv für Math. Logik* **19** (1978) 139–156.

[4] G. Pottinger, A type-assignment for the strongly normalizable λ-terms, in: J.P. Seldin and J.R. Hindley, eds., *To H.B. Curry* (Academic Press, New York, 1980) pp. 561–577.