

---

**発表概要**

---

## 多相ヴァリエントを含んだパターンマッチングの型付けについて

JACQUES GARRIGUE<sup>†</sup>

多相ヴァリエントは Objective Caml に導入されてから、よく使われるようになってきた。代数的型定義の構造による等価性を可能にし、多相性によるコードの共有・再利用も促進している。多相ヴァリエントの型付けとコンパイル方法は過去に研究され、すでに発表されている。しかしながら、多相ヴァリエントはパターンマッチングによって分解されるが、今まで浅いパターンマッチングしか考慮されてこなかった。浅いパターンマッチングに限定すると、パターンが対応している場合は自明で、型付けは対応漏れを防ぐ。深いパターンマッチングを許すと、様々な型の値が同じパターンマッチングの中で出現し、互いに影響する可能性がある。対応漏れの検索が型付けの後でしかできないので、それを型付けに利用することができないという。問題を説明し、多相ヴァリエントを含んだパターンマッチングの(対照的な)型付けアルゴリズムを提案する。

## Typing Deep Pattern-matching in Presence of Polymorphic Variants

JACQUES GARRIGUE<sup>†</sup>

Polymorphic variants are a well-known feature of the Objective Caml programming language, and they have turned popular since their introduction. They allow structural equality of algebraic type definitions, and code reuse through their polymorphism. Their typing and compilation have been studied in the past, and there are already detailed published works for both [1] and [2]. By their very nature, polymorphic variants depend on pattern matching to analyze their contents. However, only typing for shallow pattern matching was studied in the past. In that case, checking exhaustivity is trivial, and the natural typing rule guarantees it. Deep pattern matching is more complex, as other constructors may appear nested in the same pattern matching. Exhaustivity check is available, but only after finishing type checking, while we would like to use it to define the typing of polymorphic variant patterns. We explain the tradeoffs, and define a type checking algorithm for pattern-matching containing polymorphic variants which is symmetric.

[1] Jacques Garrigue: Simple Type Inference for Structural Polymorphism, FOOL9, 2002.

[2] Jacques Garrigue: Programming with polymorphic variants, ML Workshop, 1998.

(平成16年1月20日発表)

---

<sup>†</sup> 京都大学数理解析研究所  
Research Institute for Mathematical Sciences, Kyoto  
University