

---

**発表概要**

---

## Scheme における Evaluation Strategy の設計と実装

西村 祥治<sup>†</sup> 湯浅 太一<sup>†</sup>  
八杉 昌宏<sup>†</sup> 小宮 常康<sup>†</sup>

Evaluation Strategy とは、並列プログラムを記述する一手法で、アルゴリズムと動的な振舞を分けてプログラムを記述することを可能にする。これは lazy な言語である Haskell で提案されている手法である。遅延評価は評価する式を動的に決定するため、式の評価順序は不定である。しかし、評価要求を与えることによってそれを制御することは可能であり、Evaluation Strategy はこれを利用している。本発表ではこの手法を Strict な言語である Scheme に導入する。導入にあたっては、Scheme に遅延評価の仕組みが必要である。Scheme には `delay`, `force` があり、これらを用いた遅延評価を実現することは可能であるが、プログラムの至るところに埋め込む必要があり、エレガントな手法ではない。本発表では Scheme の言語機能として遅延評価系を導入し、そのインタフェースを用意することで遅延評価を容易に利用できる仕組みを提案する。それによって、Scheme 上で Evaluation Strategy を実現する。

## Design and Implementation of Evaluation Strategy on Scheme

SHOJI NISHIMURA,<sup>†</sup> TAICHI YUASA,<sup>†</sup> MASAHIRO YASUGI<sup>†</sup>  
and TSUNEYASU KOMIYA<sup>†</sup>

Evaluation Strategy is a method to describe parallel programs, which enables us to describe the dynamic behavior of a program separately from the algorithm. It is proposed for a lazy programming language, Haskell. Since lazy evaluation dynamically decides which expression should be evaluated next, the evaluation order depends on execution. However, it is controllable by giving evaluation request to expressions and Evaluation Strategy uses this gauge, Scheme. For this purpose, it is necessary to introduce lazy evaluation mechanism into Scheme. Since Scheme has `delay` and `force`, we could construct lazy evaluation by using them. But we have to insert `delay` and `force` all over a program and it is not an elegant way. In this presentation, we propose to introduce lazy evaluator into Scheme and to provide `lazy` and `touch` as the interface to use it easily. Using this mechanism, we implement Evaluation Strategy on Scheme.

(平成 13 年 1 月 23 日発表)

---

<sup>†</sup> 京都大学情報学研究科  
Graduate School of Informatics, Kyoto University