

発表要旨

関数型言語SASL+LVに、GHCの"コミット"の概念を採り入れた言語について解説する。

この言語は、形式引数のユニフィケーションを行うガード部と、ゴール部として用いる実引数を仕様として備えている。このインプリメンテーションは、コンビネータS, K, Iによって行われる。また、コンパイルーションは、古典的なターナーの方法の枠組みで行われる。

SASL+LVに対する拡張は、次の三点である。すなわち、

1. ガード部を、明示的に示す記号（形式引数を実引数と区別する為の記号）の導入
2. コミットするまでの、競争状態を制御する、solveというライブラリ関数の導入
3. 実引数をユニファイしてしまわないよう、プロテクトするコンビネータRの導入

このインプリメンテーションを通じて、正規順序評価と、ユニフィケーションの組合せは非常に強力であるが、取り扱いに注意を要するということが分かった。

質疑応答

質問：より高階の、関数のユニフィケーションはどの様に解決したのか。

回答：関数型言語に一般的に生じるユニフィケーションの問題はここでは解決されていない。

ここでとった方法はごく普通のもので、状況は、良くも悪くもなっていない。したがって関数同士のユニフィケーション、すなわち、関数の等価性の問題などは解決されていない。

また、カーチェックはここでは行っていない。なぜならそれは、しばしば有用であるところの連続する構造を作ることが妨げるからだ。オカーチェックをすると、 $x$ と $x+1$ を $=$ で結んだ時、困ることが起きるが、ここではユニファイで事を選んでるので、サスペンドすることはない。