

プログラミング言語Egison 中間発表

萩谷研究室
江木聡志

プログラミング言語Egison

- 純粹関数型言語
- 超強力なデータデコンストラクト機能
- インタプリタをHaskellで実装している
- Hackageのパッケージとして配布している

Egisonを開発した動機

- 数学のアルゴリズムをより自然なプログラムとして表現したい
- 既存のプログラミング言語では集合やマルチセットの扱いが自然に扱えない
 - 集合やマルチセットなどを扱う際、いちいちリストとして変換して扱わなければならない

既存のパターンマッチの問題点

- 正規形を持つデータについては、正規形に変換して、パターンマッチが行える
- 集合やマルチセットのような正規形を持たないデータに対してはパターンマッチが難しい
- 正規形を持たないデータに対してのパターンマッチには探索が必要であることがその原因

主な先行研究

- P. Wadler. Views: A way for pattern matching to cohabit with data abstraction. In Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages, page 313. ACM, 1987.
- M. Erwig. Active patterns. Implementation of Functional Languages, pages 21–40, 1996.
- M. Tullsen. First Class Patterns. Practical Aspects of Declarative Languages, pages 1–15, 2000.
- S. Antoy. Programming with narrowing: A tutorial. Journal of Symbolic Computation, 45(5):501– 522, 2010.
- ...
- S. Egi. Pattern Matching for Unfree Data Types, JSSST 2011, Japan Society for Software Science and Technology, 2011.

Egisonの目標

- 人間が頭の中で当たり前に行えるパターンマッチの操作を簡潔に表現できるようにすること
- 特にマルチセットや集合のようなコレクション型についてのパターンマッチを簡潔に表現できるようにする

Egisonのとした解決案

- パターンとデータを引数に取り, 全ての可能な束縛を計算するマッチ関数なるものを型ごとにユーザは記述する
 - 例. Multisetのマッチ関数は, パターン`<cons $x $xs>`, ターゲット`{1 2 3}`の場合, 以下のような結果を返す
 - `{[x 1] [xs {2 3}]}, {[x 2] [xs {1 3}]},`
`{[x 2] [xs {1 3}]}`
- マッチ関数の記述のために以下の3つの関数を定義する
 - Var-match: パターンが変数パターンである場合にパターンマッチを行う関数
 - Inductive-match: パターンが帰納的に構成されたパターンである場合にパターンマッチを行う関数
 - Equal?: パターンが値である場合にパターンマッチを行う関数

表現を簡略化するための工夫

- どの型でパターンマッチを行うのかを指定してパターンマッチを行う
- 同じパターンの内の変数に束縛された値を他の部分をパターンマッチする際に参照できる
- パターンマッチの際に行われる探索を簡潔に制御する仕組みがある

Egisonの貢献

- パターンの表現力が向上したことにより、今まで簡潔に表現できなかったパターンが簡潔に表現できるようになった
- 特にコレクションのパターンマッチについては直感のままに表現できるようになった
 - 例. ポーカーの役を判定するパターンマッチ

今後の方針

- コンパイラや豊富なライブラリを作り、もっと実用に耐えうる言語を目指す
- パターンマッチのさらなる拡張
 - 連想配列のようなデータ構造に対しても自然なパターンマッチを実現する