演習 3 (5/21)

31010 佐藤秀明

概要

- XSLT と CDuce の比較
- Cduce の仕様 / 実装

参考文献

- XSL Transformations (XSLT) Version 1.0 W3C Recommendation 16 November 1999
  - http://www.w3.org/TR/xslt
- CDuce: An XML-Centric General-Purpose Language
  - http://www.cduce.org/papers/cduce-design.ps.gz

XSLTとCDuceの比較(1)

- XML形式 vs OCaml風文法
  - 一般に、タグは記述量が多い。
    - タグは最後に閉じなければならない
    - “タグ + 属性”が 関数適用に相当
  - 一般に、関数型言語は記述量が少ない。
    - 一度定義した関数を繰り返して使用
    - 関数引数に未評価の expression を含むことができる

XSLT と CDuce の比較 (2)

- 変換ルールの適用先指定
  - XSLT では、
    - 提示された変換ルールの羅列からマッチするルールを xsl:apply-templates で自動的に選択することが可能
    - 現在のノード下の同種な子ノードを一括して apply-templates できる
  - CDuce ではどの引数をどの関数に適用させるかは明示的に指定する必要がある
    - 関数のオーバーロードと map で代用可能

XSLT と CDuce の比較 (3)

- 型の導入 (CDuce)
  - 読み込んだ XML 構造を型チェックできる
  - 型に対する正規表現
    - * や ? を用いて複雑な型の変数 bind を一括実行
  - パターンマッチの静的な最適化

CDuce の型システム (1)

- 型 = 値の集合
  - 値そのものも型である
    - 例 :1 は「 1 」という型である (「 Int 」でもある)
  - subtype の包含関係に関して健全かつ完全
  - 型に対するシンプルな集合論的演算
    - & 、| 、\ など
    - 例 :Int\1　… 1 以外の整数
    - 例 :Int|String　…整数または文字列

CDuce の型システム (2)

- パターンマッチ
  - 型の正規表現を利用
    - 例 :x&Int　… Int 型であるものを x に bind
    - 例 :(i::Int|s::String|b::Bool)*　…シーケンスの中で Int 、String 、Bool であるものをそれぞれまとめて i 、s 、b に bind
  - 再帰的なパターンマッチ
    - P where P =(d & Char, Q)|(_, P)
      and Q = (d & Char, Q)|(d&'.', (d & Char, R))
      and R = (d & Char, R)|(d & `nil)

CDuce の型システム (3)

- 関数のオーバーロード
  - 例 :let f (Int->Int; Strng->String)
    x & Int -> x
    | x & String -> x
  - 引数の型に応じた操作を行う

CDuce の型システム (4)

- パターンマッチの最適化
  - 例 :type A = <x>[...]
    type B = <y>[...]
    let f (<a>[A+|B+] -> Int)
      <a>[A+] -> 0
    | <a>[B+] -> 1
  - 最適化後 : <_>[<x>_] -> 0
    | _ -> 1
  - 可読性を優先したパターンの記述を、コンパイル時に処理系が効率を優先したパターンに変換する