Context Free Expressions

Context Free Expressions

- Just as the regular languages over an alphabet have a convenient shorthand (regular expressions) the context free languages also have a convenient short hand, context free expressions.

- **Definition.** The set of *context free expressions* (with respect to Σ) is defined inductively by the following rules:
  1. The symbols $\emptyset$ and Λ are context free expressions
  2. Every symbol $\alpha \in \Sigma$ is a context free expression
  3. If E and F are context free expressions, then (Mu @i . E), (EF) and (E+F) are regular expressions.
  4. In a surrounding Mu context, @i , is a context free expression.

Definition as a datatype

```
• data CfExp a
•   = Lambda                          -- the empty string “”
•   | Empty                           -- the empty set {}
•   | One a                           -- a singleton set {a}
•   | Union (CfExp a) (CfExp a)       -- union of two CfExp
•   | Cat (CfExp a) (CfExp a)         -- Concatenation
•   | Mu Int (CfExp a)                -- Recursion
•   | V Int                           -- Variable
```

- Note the two new kinds of expressions
- Mu and V, which replace the Star operator of the regular expressions

Context Free Expressions as Languages

- **Definition.** For every context free expression E, there is an associated language *L(E)*, defined inductively as follows:

  1. L($\varnothing$)=$\varnothing$ and L(Λ)={Λ}
  2. L(a)={'a'}
  3. Inductive cases
     1. L(EF) = L(E) L(F)   recall implicit use of dot L(E) $\bullet$ L(F)
     2. L(E+F) = L(E) $\cup$ L(F)
     3. L(Mu @i . E) = L(E[Mu @i .E / @i])
        1. Where E[d/@i] denotes substitution of d for @i

  –

- **Definition.** A language is *context free* if it is of the form L(E) for some context free expression E.

Conventions

- We use juxtaposition to denote concatenation
- We use parentheses to denote grouping
- We use # for $\emptyset$
- We use ^ for Λ
- We denote variables as @i for all positive integers i.
- We use the (RegLang) Star operator as a shorthand
- `exp*   = Mu @i.(exp + ^)@i`

Find Context Free Expression for these languages

- $\{a^n\, b\, a^n \mid n \in Nat\}$
- $\{ w \mid w \in \{a,b\}^*,$ and w is a palindrome of even length $\}$
- $\{a^n\, b^k \mid n,k \in Nat, n \le k\}$
- $\{a^n\, b^k \mid n,k \in Nat, n \ge k\}$
- $\{ w \mid w \in \{a,b\}^*,$ w has equal number of a's and b's $\}$

Meaning of a CFExp

- `-- extract an "nth" approximation of the set denoted`

- `meaning:: Ord a => Int -> (CfExp a) -> Set [a]`
- `meaning n (One x) = one x`
- `meaning n Lambda = lam`
- `meaning n Empty = empty`
- `meaning n (Union x y) =`
- `   union (meaning n x) (meaning n y)`
- `meaning n (Cat x y) = cat (meaning n x) (meaning n y)`
- `meaning n (V m) =`
- `   error ("Unbound variable: v"++show m++".")`
- `meaning n (x@(Mu v body)) =`
- `   meaning n (unfold n v body body)`