On the Design of a *Periodic* Table of VDM Specifications

Mário Antunes

ATNoG@IT Aveiro

June 22, 2012

Outline

Motivation

- Where is the borderline between invention and sheer routine-work in formal modelling?
- To answer this it is necessary a repository of specifications
- By factoring standard algorithms it is possible to identify elementary algorithm components
- The paper presents a analysis and classification of standard algorithm in a tabular structure (Periodic Table)

Motivation: factorization versus calculation

Consider the following equation

$$x = \frac{756}{792}$$

Solving it by calculation, with a brute force approach, the result is

$$x = \frac{756}{792} = 0.95454545...$$

Instead it is possible to represent each value of the fraction with a prime factorization

$$756 = 2^2 \times 3^3 \times 7$$
$$792 = 2^3 \times 3^2 \times 11$$

Now the equation can be solved

$$x = \frac{756}{792} = \frac{2^2 \times 3^3 \times 7}{2^3 \times 3^2 \times 11} = 2^2 \times 2^{-3} \times 3^3 \times 3^{-2} \times 7 \times 11^{-1} = \frac{21}{22}$$

Divide-and-Conquer pattern

- Computing is related to problem solving
- General strategy to solve a problem is to divide it into tractable pieces

Merge Sort

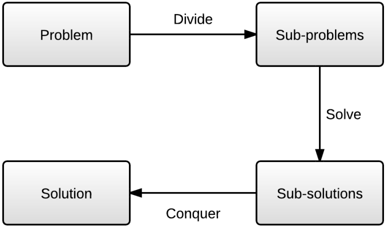
```
void mergesort(int low, int high) {
    if (low < high) {
        int middle = (low + high) / 2;
        mergesort(low, middle);
        mergesort(middle + 1, high);
        merge(low, middle, high);
    }
}
```

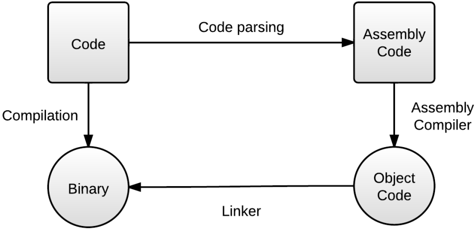
Insertion Sort

```
void insertionSort(int[] arr) {
    int i, j, newValue;
    for (i=1; i < arr.length; i++) {
        newValue = arr[i];
        j = i;
        while (j>0 && arr[j - 1]
                > newValue) {
            arr[j] = arr[j - 1];
            j--;
        }
        arr[j] = newValue;
    }
}
```

Divide-and-Conquer pattern

- Computing is related to problem solving
- General strategy to solve a problem is to divide it into tractable pieces





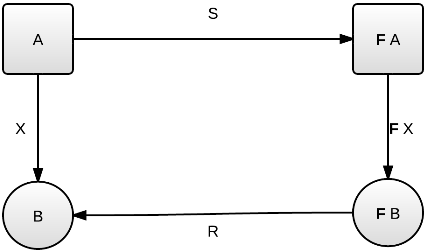
Relation approach to Divide-and-Conquer

- The Divide-and-Conquer pattern can be modelled in relation algebra

  S: divide step

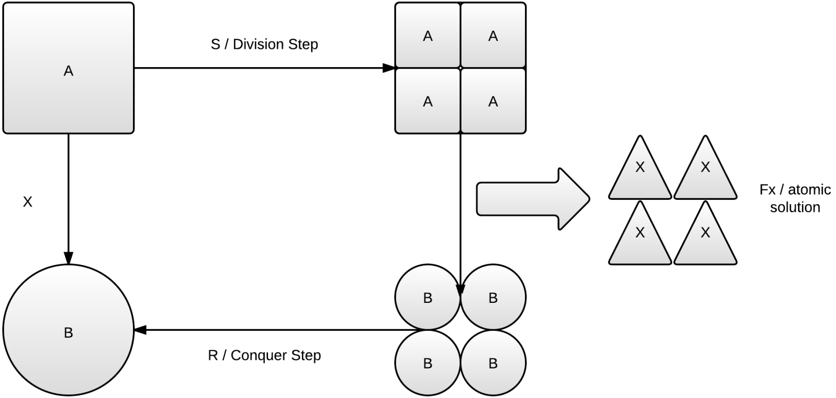
  FX: solve step / atomic solution step

  R: conquer step



$$X = R \cdot (FX) \cdot S \quad (1)$$

Relation approach to Divide-and-Conquer



Relation approach to Divide-and-Conquer

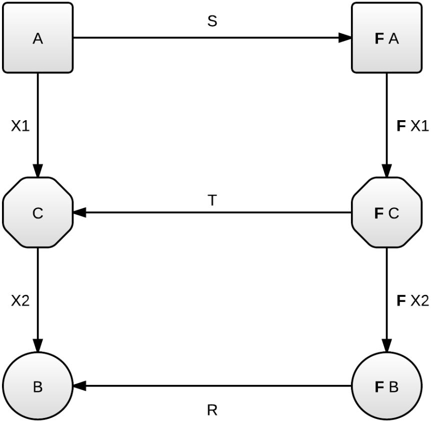
$$X = R \cdot (FX) \cdot S$$

Assuming that the algorithm can be divided into multiple steps

$$X = X1 \cdot X2$$

The model can be expanded with the following algebra

$$C \xleftarrow{T} FC$$



Relation approach to Divide-and-Conquer

S must be "well-founded" and T must be bijective over the data type defined by F

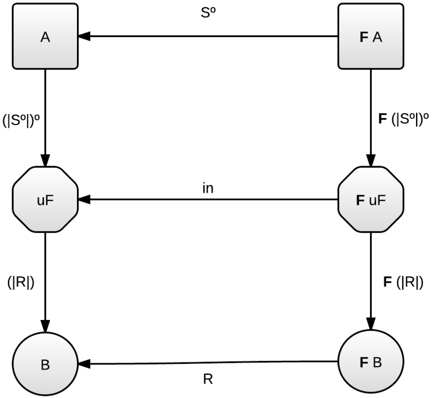
$$C = \mu F$$

μF is the virtual structure used internally by the algorithm

$$\mu F \xleftarrow{in} F(\mu F)$$

Applying this the solutions in terms of $(R \cdot FX \cdot S)$ is

$$\mu X.(R \cdot FX \cdot S) = (|R|) \cdot (|S^\circ|)^\circ$$



Example: merge sort

```
mergeSort : seq of int -> seq of int
mergeSort (l) ==
    cases l :
      [] -> l,  -- | not considered in the example
      [e] -> l, -- | represented by singl
      other -> let l1 ^ l2 in set(l)
                   be st abs(len l1 -len l2) < 2
                   -- | represented by pconc°
               in let l_l = mergeSort(l1)
                      l_r = mergeSort(l2)
                  in lmerge (l_l, l_r)
    end;
```

Example: merge sort

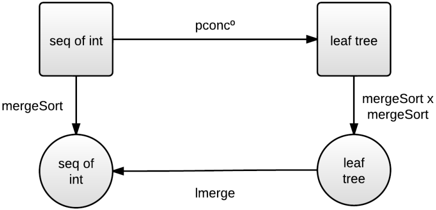
Based on the expression $X = R \cdot (FX) \cdot S$ the merge sort algorithm can be defined as:

$$mergeSort = [singl, lmerge \cdot (mergeSort \times mergeSort)] \cdot S$$
$$mergeSort = [singl, lmerge] \cdot (id + mergeSort \times mergeSort) \cdot S$$

Where it is possible to identify

$$R = [singl, lmerge]$$
$$Ff = id + f \times f$$
$$FX = int + X \times X$$

(represents a Leaf Tree)



S is defined by extracting "what remains"

$$S = [singl, pconc]^{\circ}$$

Conclusion

- The paper presents a classification of several standard algorithm based on their implicit virtual structure
- The paper also presents a second table, a formal classification of *divide* and *conquer*

Conclusion

$\mathsf{F}\,X$	$1+X$	$1+A\times X$	$A+X^2$	$1+A\times X^2$	$(B\times A+B\times X)^*$
$\mu\mathsf{F}$	nat	seq of A	$LTree$	$BTree$	$HTree$
$In \rightarrow Out$	**Specifications**				
$nat \rightarrow bool$	odd $even$				
$nat \rightarrow nat$		$square$ $factorial$	$fibonnaci$ $doubleFactorial$		
$nat \rightarrow$ set of $nat1$		$inseg$			
seq of $A \rightarrow$ seq of A		$insertSort$ $invSeq$	$mergeSort$	$quickSort$	
seq of $A \rightarrow bool$		$ordSeq$			
seq of $A \rightarrow$ set of A		$elems$			
seq of $A \rightarrow$ set of $nat1$		$inds$			
seq of $A \rightarrow$ Bag of A		$seq2bag$			
seq of $A \rightarrow nat$		len			
$LTree \rightarrow bool$			$balLTree$		
$LTree \rightarrow nat$			$depthLTree$		
$LTree \rightarrow int$			$addLTree$ $countLTree$		
$LTree \rightarrow$ seq of A			$tips$		
$LTree \rightarrow LTree$			$invLTree$		
$BTree \rightarrow bool$				$ordBTree$ $balBTree$	
$BTree \rightarrow nat$				$depthBTree$	
$BTree \rightarrow$ seq of A				$preOrder$ $inOrder$ $postOrder$	
$BTree \rightarrow$ seq of seq of A				$traces$	
$BTree \rightarrow BTree$				$invBTree$	
set of $A \rightarrow nat$		$card$			
set of $A \rightarrow$ seq of A		$Set2seq$			
set of $bool \rightarrow bool$		$\forall$ $\exists$			
set of set of $A \rightarrow$ set of A		$dunion$			
map A to $B \rightarrow$ set of A		dom			
map A to $B \rightarrow$ set of B		ran			
set of (map A to B) $\rightarrow$ map A to B		$merge$			
$PTree \rightarrow$ Bag of A					$explode$
$FS \rightarrow$ map $String$ to A					tar
$(Other)$				$hanoi$	

Conclusion

$\mathsf{F}\,X$	$1+X$	$1+A\times X$	$A+X^2$	$1+A\times X^2$	$(B\times A+B\times X)^\star$
$\mu\mathsf{F}$	nat	$\texttt{seq of}\ A$	$LTree$	$BTree$	$HTree$
Carrier	**F-(co)algebras**				
$bool$	$[\underline{\mathbf{F}},\neg]$ $[\underline{\mathbf{T}},\neg]$	$[\underline{\mathbf{F}},\vee]$ $[\underline{\mathbf{T}},\wedge]$	$[\underline{\mathbf{F}},\vee]$ $[\underline{\mathbf{T}},\wedge]$		
nat	$[\underline{0},suc]$	$[\underline{0},+]$ $odds^\circ$ $[\underline{1},*]$ $nats^\circ$	$[id,+]$ $[id,*]$ $[\underline{1},+]$ $fibd^\circ$	$[\underline{1},bmul]$ $[\underline{0},badd]$	
$nat * nat$			$dfacd^\circ$		
$\texttt{seq of}\ A$		$[\underline{[\,]},cons]$ $[\underline{[\,]},rcons]$ $[\underline{[\,]},\hat{}\,]$	$[singl,\hat{}\,]$ $[singl,pconc]$ $[singl,lmerge]$	$[\underline{[\,]},inord]$ $[\underline{[\,]},pinord]$ $[\underline{[\,]},prord]$ $[\underline{[\,]},psord]$	
$LTree$			in $in\cdot(id+sw)$		
$BTree$				in $in\cdot(id+id\times sw)$	
$HTree$					in
$\texttt{set of}\ A$		ins $pins$ $[\underline{\emptyset},\cup]$ $ins\cdot(id+\pi_1\times id)$ $ins\cdot(id+\pi_2\times id)$	$[\lambda x.\{x\},\cup]$	$[\underline{\emptyset},bputs]$	
$\texttt{map}\ A\ \texttt{to}\ B$		ins $pins$ $[\underline{\{\mapsto\}},munion]$			
$PTree$					$exsplit^\circ$
$\texttt{Bag of}\ A$					$exjoin$
$FileS$					$tsplit^\circ$
$\texttt{map}\ String\ \texttt{to}\ A$					$tjoin$
$(Other)$				$hsplit^\circ$	

End

Questions.