

Prova com consulta. Duração: 90 minutos.

Nome do estudante: \_\_\_\_\_ N° \_\_\_\_\_

1. [2 valores] Quais os resultados das seguintes expressões em VDM?

a) dunion power {2,4}

{2,4}

b) {1,2,3,4,5} psubset (inds (conc [[2,4,8,16],[32]]))

false

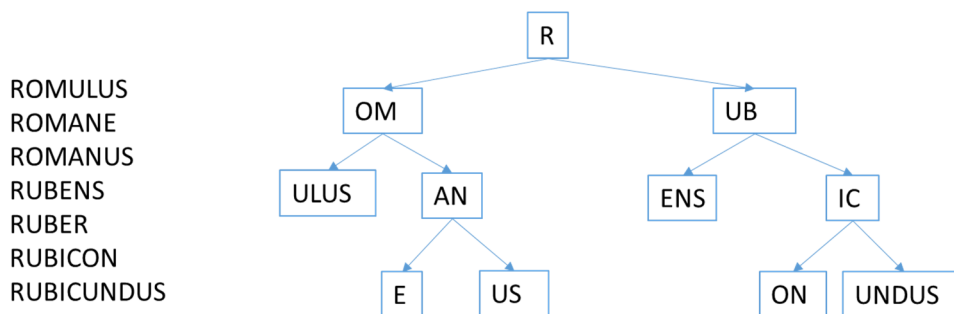
c) [1,2,3] ++ {1 |-> 0}

[0,2,3]

d) {<PT> |-> 4, <EN> |-> 2, <SP> |-> 3} munion {<PT> |-> 4, <GE> |-> 5}

{ <EN> |-> 2, <GE> |-> 5, <PT> |-> 4, <SP> |-> 3 }

2. A árvore *radix* é uma estrutura de dados do tipo árvore ordenada que pode servir para guardar palavras tal com ilustra a figura seguinte. Estas estruturas são úteis para dar suporte à escrita inteligente presente habitualmente nos *smartphones*. Quando um nó “A” da árvore tem apenas um filho “B”, os nós são compactados num só nó “AB”.



Considere a seguinte formalização em VDM++ para representar árvores *radix*.

```
class Radix
  types
    public Node = seq of char;

    public Relation :: source: Node
                    dest: Node;

  instance variables
    arrows: set of Relation;
    root: Node;

  -- inv1: "there are no nodes with only one child"
end Radix
```

Prova com consulta. Duração: 90 minutos.

a) [3 valores] Formalize o inv1 “não existem nós com um único filho”.

```
inv forall r1 in set arrows &
  exists r2 in set arrows &
    r1.source = r2.source and r1.dest <> r2.dest;
```

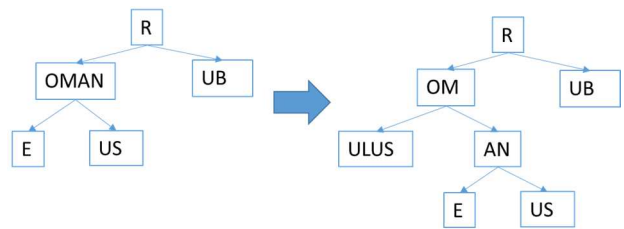
b) [3 valores] Formalize uma operação que retorne o conjunto de todos os nós de uma Radix.

```
public setofnodes: () ==> set of seq of char
setofnodes () ==
  return {n.source | n in set arrows} union {n.dest | n in set arrows};
```

c) [3 valores] Escreva uma operação que insira uma nova Relation numa árvore Radix. Escreva a condição necessária para o bom funcionamento dessa operação

```
public insert: seq of char * seq of char ==> ()
insert (n,m) ==
  arrows:= arrows union {mk_Relation(n,m)}
pre exists a in set arrows & a.source = n;
```

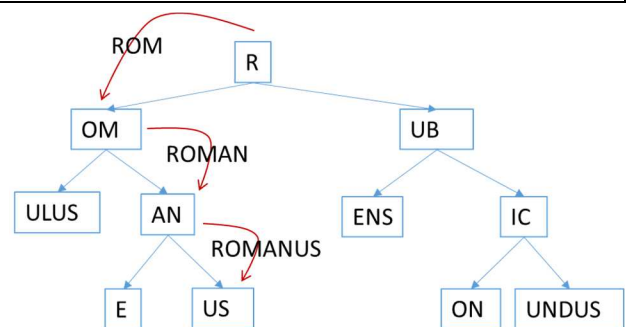
d) [3 valores] Especifique uma operação que adicione uma nova palavra a uma Radix. Esta função recebe o nó a separar, o prefixo, o sufixo e a palavra a inserir. Considerando a figura a seguir, a função seria invocada da seguinte forma:



insert(“OMAN”, “OM”, “AN”, “ROMULUS”)

```
public adicionar: seq of char * seq of char * seq of char * seq of char ==> ()
adicionar (p,s1,s2,s) ==
(
  insert(s1,s2);
  for all a in set nos(p) do
    insert(s2,a);
  let i,j in set inds s be st s(i,...,j)=s1 in
    insert(s1,s(j+1,...,len s));
  for all a in set nos(p) do
    arrows := arrows\{mk_Relation(p,a)};
);
```

e) [2 valores] Especifique uma operação que determine se uma string s existe numa árvore Radix. Por exemplo, na árvore em baixo, tanto a pesquisa da string “ROMAN” como da string “ROMANUS” dariam resultado true. A pesquisa de “RA” daria false.



(Assuma que existe uma operação subs que recebe duas strings s1 e s2 e retorna true caso s1 esteja no início de s2, por exemplo, subs(“abc”, “abcd”)=true, subs(“a”, “a”)=true e subs(“b”, “abc”)=false.)

Prova com consulta. Duração: 90 minutos.

```
public strexists: seq of char ==> bool
strexists (word) == (
  return exists s:seq of Node & conc s = word and s(1) = root and
    forall i,j in set inds s & j=i+1 => mk_Relation(s(i), s(j)) in set arrows;
);
```

3. [4 valores]

- a) Determine (através do cálculo da wp) se o seguinte triplo de Hoare é verdadeiro  
 $\{ i = 3j \}$  if  $i > j$  then  $m := i - j$  else  $m := j - i$   $\{ m - 2j = 0 \}$

```
[i>j and wp(m:=i-j;m-2j=0)] or [i<=j and wp(m:=j-i;m-2j=0)]
(i>j and i-j-2j=0) or (j<=i and j-i-2j=0)
(i>j and i-3j=0) or (j<=i and -i-j=0)
(i>j and 3j=i) or (j<=i and j=-i)
3j=i or j=-i
i = 3j => 3j=i or j=-i
True
```

O triplo é verdadeiro (para números naturais)

- b) Indique a *weakest precondition* (wp) do seguinte programa

```
{wp} while (m ≤ n) do n := n - m { ∃ k, k * A + n = B ∧ n < A }
```

$m=A \wedge n=B$  (para números naturais)

- c) Considere as seguintes afirmações:

- i. dado o triplo de Hoare  $\{S\} P \{Q\}$ , em que  $S$  é a *weakest precondition*, posso assumir que o triplo  $\{R\} P \{Q\}$  também é verdadeiro. Qual é a relação existente entre  $S$  e  $R$ ?

$R \Rightarrow S$

Prova com consulta. Duração: 90 minutos.

---

- ii. dado o triplo de Hoare  $\{S\} P \{Q\}$ , em que  $Q$  é a *strongest postcondition*, posso assumir que o triplo  $\{S\} P \{R\}$  também é verdadeiro. Qual é a relação existente entre  $Q$  e  $R$ ?

$Q \Rightarrow R$

Boa sorte!