FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Mestrado Integrado em Engenharia Informática e Computação

Métodos Formais em Eng.ª de Software

2008/2009

EXERCÍCIOS

1 – sets

1.1. What is the value of the following expressions:

1. {1,...,6}
2. {1,...,1}
3. {4,...,1}
4. {x | x in set {2,3,4,5} & x > 2}
5. {x | x in set {2,3,4,5} & 22 < x}
6. {} in set power {1,3,6}
7. dunion {{1,2},{1,5,6},{3,4,6}}
8. dinter {{1,2},{1,5,6},{3,4,6}}
9. {1,2,3} psubset {1,2}

1.2 Write down the following English language statements in VDM++

1. The size of set s is less than 5.
2. The sets s1 and s2 are non-empty and disjoint.
3. The set formed by the union of s1 and s2 will have a size smaller than 10.
4. The elements which are in both s1 and s2 will be contained in s3.

1.3 Define a predicate is disjoint which tests whether two sets are disjoint, i.e., whether they have no common elements. Hence, *IsDisjoint*({1,2,3},{3,4,5}) must evaluate to false.

1.4 Define a predicate *IsMutDisj* which tests whether the elements of a set of sets are mutually (pairwise) disjoint. The predicate must yield false on {{1,2,3},{6,7,8},{4,5,6}} and true on {{1,3,7},{2,6,8},{4,5}}.

2 – sequences

2.1. Which of the following expressions are true?

1. 6 in set elems [3,6,8,10,0]
2. [] = tl [4]
3. 6 in set inds [3,6,8,10,0]

2.2 What are the results of the following expressions:

1. tl [1,2,3]
2. len [[1,2],[1,2,3]]
3. hd [[1,2],[1,2,3]]
4. tl [[1,2],[1,2,3]]
5. elems [1,2,2,3,3,4]
6. elems [[1,2],[2],[3],[3],[3,4]]

2.3 What is the value of the following expressions

1. len []
2. len [1,2,3] + len [3]
3. [hd [<A>,<B>]] ^ [hd [<C>,<D>]]
4. tl [1,2,3,4,5] ^ [hd [1,2,2]]
5. tl ([1,2]^[1,2])

2.4 Write down the following English language statements in VDM++:

1. The length of the sequence *s* is less than l4.
2. The first three elements of the sequence *p* are subset of the set {2,4,6,8}.
3. The middle element of the sequence *s*, which contains 9 elements, is less than 421.
4. The final element of the sequence *s1* is greater than 405.
5. The length of the sequence formed by concatenating the sequence *s2* and *s4* is greater than 15.
6. The element <ALF> will never be found in the sequence *s*.
7. The sequence *prior* will always contain a non-empty subset of the first five elements of the sequence *totalpr*.

3 – maps

3.1. What is the value of the following expressions?

1. dom {100 |-> <TIM>, 10 |-> <ROB>, 12 |-> <DAVE>}
2. rng {100 |-> <TIM>, 10 |-> <ROB>, 12 |-> <DAVE>}
3. {1000 |-> 3, 1005 |-> 4, 1002 |-> 1} ++ {1002 |-> 6}
4. {1008 |-> 3, 1065 |-> 4, 1012 |-> 1} ++ {1011 |-> 6}
5. {128} <: {100 |-> <TIM>, 10 |-> <ROB>, 12 |-> <DAVE>}
6. {128} <-: {100 |-> <TIM>, 10 |-> <ROB>, 12 |-> <DAVE>}

3.2. Assume a map price which relates cars to their price. Furthermore, assume that the set BL contains the cars made by British Leyland and that the set Fiat contains the cars made by Fiat. Write down the following descriptions in VDM++.

1. No car is made by both British Leyland and Fiat.
2. The price of the Montego 1.3L (a car made by British Leyland).
3. The prices of all Fiat cars.
4. The number of Fiat cars that have a price between £6000 and £7000.
5. The map formed by changing the price of the Montego 1.3L to £5500.

3.3. If the map accounts relates the name of the accounts holders to the set of accounts that they own and the map balance relates account names to their balance, then write down a VDM++ expression that equals:

1. The number of overdrawn accounts.
2. The number of accounts owned by Wilkinson.
3. The set of account holders who only have one account.
4. The set of names of account holders who have at least one account that is overdrawn.
5. The total balance of all accounts owned by Roberts.

4 – maps and sequences

4.1. For each informal description below you must write the corresponding VDM++ expression and its type:

1. Describe the map which maps the natural numbers less than 5 to themselves. This identity map construction should be made by means of both map enumeration and map comprehension.
2. Describe the map which maps all natural numbers less than 5 to the character 'a'. The map construction should be made by means of both map enumeration and map comprehension.
3. Describe the map which maps the even natural number less than 5 to the double of their own value. The map construction should be made by means of both map enumeration and map comprehension.
4. Describe the sequence of pairs where the first element corresponds to the index in the sequence and the second element is true. The length of the sequence should be 4. The sequence construction should be made by means of both sequence enumeration and sequence comprehension.
5. Describe a sequence of length 5 whose elements are records tagged by A. Each record must contain a singleton set whose element is the index of the record in the sequence. The sequence construction should be made by means of both sequence enumeration and sequence comprehension.
6. Describe a set of sets of Booleans. The construction should be described by means of both set enumeration and the power set constructor (perhaps used in connection with a set comprehension).
7. Describe a set of nonempty sequences of natural numbers less than 4 where the length of the sequences should be less than 3 and the elements in the sequences are ordered increasingly (e.g., sequences like [1,3] but not sequences like [2,1]). There should be no duplicates in the sequences. The construction should be described by means of both set enumeration and set comprehension.

5 – binding

```
cases a:
        {b, 3, 7}               -> mk_(<SETENUM>,b),
        s union {2,5}           -> mk_(<SETUNION>,s),
        [e_1, 3, e_2]           -> mk_(<SEQENUM>, e_1, e_2),
        [6,1] ^ l               -> mk_(<SEQCONC>,l),
        mk_(2,c,-)              -> mk_(<TUPLE>,c),
        mk_A (b,b,3)            -> mk_(<RECORD>,b),
        7, true, <RED>          -> <MATCHVAL>,
        c                       -> mk_(<IDENTIFIER>,c),
        others                  -> undefined
end
```

Calculate the value of this expression for each of the bindings of the identifier *a* bellow. The type of the result must also be specified.

1. a = {2,3,4,5}
2. a = [6,1,3,5]
3. a = 7
4. a = [2,3,7]
5. a = mk_B(5,5,3)
6. a = mk_(2,{1,2},[1,2,4])
7. a = mk_A({2 |-> 1}, {2 |->1}, 3)
8. a = [[1,2], 3, {7,9}]
9. a = mk_A(7,9,3)
10. a = {2,3,7}

6 – sets, sequences and maps

```
b1 = true
b2 = false
s1 = {3,7,9}
s2 = {1,5,9}
l1 = [2,6,9,3,6,2]
l2 = [1,7,7,3]
l3 = [[2,6,9,3,6,2],[1,7,7,3],5]
m1 = {'a'|->7, 'b'|->5,'c'|->9,'d'|->7}
m2 = {3|->'y',6|->'e',1|->'s'}
m3 = {'d'|->3, 'b'|->3}
```

1. not (b1 and b2)
2. (b1 or b2) => (b1 => b2)
3. (b1 and b2) <=> (not b1 and not b2)
4. 3 in set (s1 inter s2)
5. {1,3,9} in set {s1,s2}
6. {1,3,9} subset dunion {s1,s2}
7. (s1 \ {3,5,7}) = (s2 \ {1,3,5})
8. {} psubset s1
9. power s1
10. card (s1 union s2)
11. card {}
12. card power s2
13. dinter {s1,s2}
14. {a | a in set (s2 union s1) & a >2}
15. {b + 6 | b in set {1,...,10} & ({b} subset s1) or b<5}
16. hd (l1 ^ l2)
17. tl (l1 ^ l2)
18. len ([l1, l2])
19. elems (conc [l1, l2])
20. elems (tl l2)
21. inds ([l1, l2])
22. l3(2)
23. card (inds l1 union inds l2)
24. inds (l3(1))
25. [l1(i) | i in set inds l1 & i<> 3]
26. l2 = [1,3,7,7]
27. l3(1)(2)
28. dom (m1 munion m2)
29. card rng (m1 munion m2)
30. m1 ++ m3
31. dom ({'a','d'} <-: m1)
32. rng (s1 <: m2)
33. m3 = {'b' |->3, 'd' |->3}
34. merge {m1, m2}
35. {m3('d'), dom m2, m1('d')}
36. {i+5 |-> l1(i) | i in set dom (m2)}
37. {mk_(a,b) | a in set {1,...,9}, b in set {1,2} & 1 < a and a < 4}
38. [mk_((b=6),b) | b in set {4,5,6}]
39. {i |-> mk_((i<>3),7) | i in set inds l2}
40. mk_(m1('a'),card s1, b2, len l1 = 3)