MODULE *AssignmnetOne2020*

Assignment 1 *SWEN*421 2020 marks out of 100

EXTENDS *Integers*, *Sequences*, *TLC*, *FiniteSets*

Part 1 total 65 marks

(1) 5 marks Look up the online Example from Practical TLA+ (we discussed this in the lectures. This is a version of Reduce that can be applied to Sets. You can use it as a tmeplate to define some of the other functions. If you are unsure what Map, Filter, .. mean you can look up the functions in *Java*'s Stream *API*.

$Add(x, y) \triangleq x + y$ Example output
$SetReduce(Add, \{2, 4, 6, 8\}, 0) = 20$

RECURSIVE $SetReduce(\_, \_, \_)$

(2) 10 marks $D(x) \triangleq 2 * x$ Example output
$SetMap(D, \{2, 3, 4, 5\}) = \{4, 6, 8, 10\}$

RECURSIVE $SetMap(\_, \_)$

(3) 10 marks
$SetFilter(IsTriple, \{1, 2, 3, 4, 5, 6, 7, 8, 9\}) = \{3, 6, 9\}$

RECURSIVE $SetFilter(\_, \_)$

(4) 5 marks $SetFlatMap(D, \{\{1\}, \{1, 2, 4\}, \{5, 6\}\}) = \{2, 4, 8, 10, 12\}$

RECURSIVE $SetFlatMap(\_, \_)$

(5) 5 marks

RECURSIVE $SetDistinct(\_)$

(6) 10 marks BOTH recursive AND non recursive

is the first parameter a "set of sets" a partition of the second parameter
$Part(\{\{1\}, \{2\}\}, \{1, 2\}) = \text{TRUE}$, $Part(\{\{1\}, \{2, 3\}\}, \{1, 3, 2\}) = \text{TRUE}$

RECURSIVE $PartRec(\_, \_)$

$Part(Sp, S) \triangleq$

return the set of partitions of a given set

(7) 10 marks BOTH recursive AND non recursive
$FlatSet(\{\{1, 2\}, \{3\}, \{2, 4\}\}) = \{1, 2, 3, 4\}$

$FlatSet(S) \triangleq$

RECURSIVE $FlatSetRec(\_)$

(8) 10 marks BOTH recursive AND non recursive
$PartitionsRec(\{1, 2, 3\}) = \{\{\{1, 2, 3\}\}, \{\{1\}, \{2, 3\}\}, \{\{2\}, \{1, 3\}\}, \{\{3\}, \{1, 2\}\}, \{\{1\}, \{2\}, \{3\}\}\}$

RECURSIVE $PartitionsRec(\_)$

hopefully an easier non recursive solution

$Partitions(S) \triangleq$

$Add(x, y) \triangleq x + y$
$D(x) \triangleq 2 * x$
$IsTriple(x) \triangleq x\%3 = 0$

Part 2 (total 35 marks)

Define the *Fibonacci* sequence (Google it if unsure).

1. (10 marks) Define a recursive function in TLA+ that given a number of steps returns the *Fibonacci* value (see below)

2. (15 marks) Define a TLA+ State Machine using *Init*, Next and Spec that takes a parameter (instantiated when you build the model) defining the number of steps to be taken
   The State Machine must compute the *Fibonacci* series and print the result (see below).

3. (10 marks) Add invariants: a. the recursive function (1) agrees with the TLA+ State Machine (2) b. simple type checking

EXAMPLE output from the State Machine For input 10 output the following lines
"$1 \rightarrow 1$"
"$2 \rightarrow 1$"
"$3 \rightarrow 2$"
"$4 \rightarrow 3$"
"$5 \rightarrow 5$"
"$6 \rightarrow 8$"
"$7 \rightarrow 13$"
"$8 \rightarrow 21$"
"$9 \rightarrow 34$"

RECURSIVE $fib(\_)$

CONSTANT
VARIABLES

\ * Modification History
\ * Last modified *Wed Mar* 25 10:25:39 *NZDT* 2020 by *dstr*
\ * Created *Wed Feb* 20 14:27:27 *NZDT* 2019 by *dstr*