

```

1  |----- MODULE OT -----|
   | This module contains the basic OT (Operational Transformation) functions for two operations |
   | and general ones involving operation sequences. |
6  | EXTENDS OpOperators, SetUtils |
7  |-----|
8  XformII(lins, rins)  $\triangleq$  lins is transformed against rins
9      IF lins.pos < rins.pos
10         THEN lins
11         ELSE IF lins.pos > rins.pos
12             THEN [lins EXCEPT !.pos = @ + 1]
13             ELSE IF lins.ch = rins.ch
14                 THEN Nop
15                 ELSE IF lins.pr > rins.pr
16                     THEN [lins EXCEPT !.pos = @ + 1]
17                     ELSE lins
18
19  XformID(ins, del)  $\triangleq$  ins is transformed against del
20      IF ins.pos  $\leq$  del.pos
21         THEN ins
22         ELSE [ins EXCEPT !.pos = @ - 1]
23
24  XformDI(del, ins)  $\triangleq$  del is transformed against ins
25      IF del.pos < ins.pos
26         THEN del
27         ELSE [del EXCEPT !.pos = @ + 1]
28
29  XformDD(ldel, rdel)  $\triangleq$  ldel is transformed against rdel
30      IF ldel.pos < rdel.pos
31         THEN ldel
32         ELSE IF ldel.pos > rdel.pos
33             THEN [ldel EXCEPT !.pos = @ - 1]
34             ELSE Nop
35
36  Xform(lop, rop)  $\triangleq$  lop is transformed against rop
37      CASE lop = Nop  $\vee$  rop = Nop  $\rightarrow$  lop
38           $\square$  lop.type = "Ins"  $\wedge$  rop.type = "Ins"  $\rightarrow$  XformII(lop, rop)
39           $\square$  lop.type = "Ins"  $\wedge$  rop.type = "Del"  $\rightarrow$  XformID(lop, rop)
40           $\square$  lop.type = "Del"  $\wedge$  rop.type = "Ins"  $\rightarrow$  XformDI(lop, rop)
41           $\square$  lop.type = "Del"  $\wedge$  rop.type = "Del"  $\rightarrow$  XformDD(lop, rop)
42  |-----|
   | Generalized OT functions on operation sequences. |
46  RECURSIVE XformOpOps(-, -, -)
47  XformOpOps(xform(-, -), op, ops)  $\triangleq$  Transform an operation op against an operation sequence ops.
48      IF ops =  $\langle \rangle$ 
49         THEN op
50         ELSE XformOpOps(xform, xform(op, Head(ops)), Tail(ops))

```

```

52 RECURSIVE  $XformOpOpsX(-, -, -)$ 
53  $XformOpOpsX(xform(-, -), op, ops) \triangleq$  Transform an operation  $op$  against an operation sequence  $ops$ .
54 IF  $ops = \langle \rangle$ 
55 THEN  $\langle op \rangle$  Maintain and return the intermediate transformed operations.
56 ELSE  $\langle op \rangle \circ XformOpOpsX(xform, xform(op, Head(ops)), Tail(ops))$ 
57
58  $XformOpsOp(xform(-, -), ops, op) \triangleq$  Transform an operation sequence  $ops$  against an operation  $op$ .
59 LET  $opX \triangleq XformOpOpsX(xform, op, ops)$ 
60 IN  $[i \in 1 \dots Len(ops) \mapsto xform(ops[i], opX[i])]$ 
61
62 RECURSIVE  $XformOpsOps(-, -, -)$  Transforms an operation sequence  $ops1$  against another one  $ops2$ ;
63  $XformOpsOps(xform(-, -), ops1, ops2) \triangleq$  see Definition 2.13 of the paper “Imine@TCS06”.
64 IF  $ops2 = \langle \rangle$ 
65 THEN  $ops1$ 
66 ELSE  $XformOpsOps(xform, XformOpsOp(xform, ops1, Head(ops2)), Tail(ops2))$ 
67
68 \ * Modification History
69 \ * Last modified Thu Jan 03 16:43:14 CST 2019 by hengxin
70 \ * Created Sun Jun 24 15:57:48 CST 2018 by hengxin

```