MODULE OT

Specification of OT (Operational Transformation) functions. It consists of the basic OT functions for two operations and more general ones involving operation sequences.

EXTENDS Op

OT (Operational Transformation) functions.

Naming convention: I for "Ins" and D for "Del".

The left "Ins" *lins* transformed against the right "Ins" *rins*.

$$
\begin{array}{l}
XformII(lins,\ rins) \triangleq \\
\quad \text{IF } lins.pos < rins.pos \\
\quad\ \text{THEN } lins \\
\quad\ \text{ELSE IF } lins.pos > rins.pos \\
\qquad\ \text{THEN } [lins \text{ EXCEPT } !.pos = @ + 1] \\
\qquad\ \text{ELSE IF } lins.ch = rins.ch \\
\qquad\qquad\ \text{THEN } Nop \\
\qquad\qquad\ \text{ELSE IF } lins.pr > rins.pr \\
\qquad\qquad\qquad\ \text{THEN } [lins \text{ EXCEPT } !.pos = @ + 1] \\
\qquad\qquad\qquad\ \text{ELSE } lins
\end{array}
$$

The left "Ins" *ins* transformed against the right "Del" *del*.

$$
\begin{array}{l}
XformID(ins,\ del) \triangleq \\
\quad \text{IF } ins.pos \leq del.pos \\
\quad\ \text{THEN } ins \\
\quad\ \text{ELSE } [ins \text{ EXCEPT } !.pos = @ - 1]
\end{array}
$$

The left "Del" *del* transformed against the right "Ins" *ins*.

$$
\begin{array}{l}
XformDI(del,\ ins) \triangleq \\
\quad \text{IF } del.pos < ins.pos \\
\quad\ \text{THEN } del \\
\quad\ \text{ELSE } [del \text{ EXCEPT } !.pos = @ + 1]
\end{array}
$$

The left "Del" *ldel* transformed against the right "Del" *rdel*.

$$
\begin{array}{l}
XformDD(ldel,\ rdel) \triangleq \\
\quad \text{IF } ldel.pos < rdel.pos \\
\quad\ \text{THEN } ldel \\
\quad\ \text{ELSE IF } ldel.pos > rdel.pos \\
\qquad\ \text{THEN } [ldel \text{ EXCEPT } !.pos = @ - 1] \\
\qquad\ \text{ELSE } Nop
\end{array}
$$

Transform the left operation *lop* against the right operation *rop* with appropriate OT function.

$$
\begin{array}{l}
Xform(lop,\ rop) \triangleq \\
\quad \text{CASE } lop = Nop \lor rop = Nop \to lop \\
\qquad \square\ lop.type = \text{"Ins"} \land rop.type = \text{"Ins"} \to XformII(lop,\ rop)
\end{array}
$$

```
        □ lop.type = "Ins" ∧ rop.type = "Del" → XformID(lop, rop)
        □ lop.type = "Del" ∧ rop.type = "Ins" → XformDI(lop, rop)
        □ lop.type = "Del" ∧ rop.type = "Del" → XformDD(lop, rop)
```

Generalized *OT* functions on operation sequences.

Iteratively/recursively transforms the operation *op* against an operation sequence *ops*.

```
RECURSIVE XformOpOps(_, _)
XformOpOps(op, ops) ≜
    IF ops = ⟨⟩
        THEN op
        ELSE XformOpOps(Xform(op, Head(ops)), Tail(ops))
```

Iteratively/recursively transforms the operation *op* against an operation sequence *ops*. Being different from *XformOpOps*, *XformOpOpsX* maintains the intermediate transformed operation

```
RECURSIVE XformOpOpsX(_, _)
XformOpOpsX(op, ops) ≜
    IF ops = ⟨⟩
        THEN ⟨op⟩
        ELSE ⟨op⟩ ∘ XformOpOpsX(Xform(op, Head(ops)), Tail(ops))
```

Iteratively/recursively transforms the operation sequence *ops* against an operation *op*.

```
XformOpsOp(ops, op) ≜
    LET opX ≜ XformOpOpsX(op, ops)
    IN  [i ∈ 1 .. Len(ops) ↦ Xform(ops[i], opX[i])]
```

Iteratively/recursively transforms an operation sequence *ops*1 against another operation sequence *ops*2.

See also Definition 2.13 of the paper "Imine @ *TCS*06".

```
RECURSIVE XformOpsOps(_, _)
XformOpsOps(ops1, ops2) ≜
    IF ops2 = ⟨⟩
     THEN ops1
     ELSE XformOpsOps(XformOpsOp(ops1, Head(ops2)), Tail(ops2))
```

* Modification History
* *Last* modified Sat *Jul* 07 13:35:17 *CST* 2018 by *hengxin*
* Created Sun *Jun* 24 15:57:48 *CST* 2018 by *hengxin*