```
┌──────────────────── MODULE Commons ────────────────────┐
EXTENDS Integers, Sequences
VARIABLES msgs, fstates, thrds, waitForMsgs, killed, seq
CONSTANTS PLACE, MXFINISHES, PROG_HOME, BACKUP
├─────────────────────────────────────────────────────────┤
ROOT_FINISH ≜ "distroot"

REMOTE_FINISH ≜ "distremote"

MXTHREADS ≜ 2
MXACTIVITIES ≜ 20
MXMESSAGES ≜ 200

MXFID ≜ MXFINISHES + 1

NotID    ≜ −1

NoParent    ≜ 0

FIRST_ID ≜ 1

PIDRange  ≜ NoParent . . MXFID

IDRange ≜ FIRST_ID . . MXFID

NotPlace  ≜ CHOOSE v : v ∉ PLACE

ThreadID ≜ 0 . . MXTHREADS − 1

NotThreadID          ≜ −5050

EMPTY_BLOCK ≜ −1

BlockID ≜ 0 . . 25  NBLOCKS − 1

NotBlockID ≜  −1000

StmtID ≜ 0 . . 5  MXSTMTS − 1

I_START ≜  −1

I_PRE_FIN_ALLOC ≜  −2

├─────────────────────────────────────────────────────────┤
Record Types
Sequences ≜ [aseq : 1 . . MXACTIVITIES, mseq : 1 . . MXMESSAGES, fseq : IDRange]

 Each thread has a stack, and this is the stack entry
StackEntry ≜ [b : BlockID,                    BlockID ≜ 0 . . NBLOCKS − 1
              i : StmtID ∪ {−1, −2},          StmtID ≜ 0 . . MXSTMTS − 1
              fid : PIDRange]
```

the processing unit of program instructions

$$
\begin{aligned}
\mathit{Thread} \triangleq\ [&\mathit{tid} : \mathit{ThreadID},\\
&\mathit{status} : \{\text{“idle”}, \text{“running”}, \text{“blocked”}\},\\
&\mathit{blockingType} : \{\text{“NA”}, \text{“FinishEnd”}, \text{“AsyncTransit”}, \text{“FinishAlloc”}, \text{“AsyncTerm”}\},\\
&\mathit{stack} : \mathit{Seq}(\mathit{StackEntry})]
\end{aligned}
$$

the activities that are pushed to scheduler's ready queue,
and will eventually be fetched by threads

$$
\begin{aligned}
\mathit{Activity} \triangleq\ [&\mathit{aid} : \mathit{Nat},\\
&b : \mathit{BlockID},\\
&\mathit{fid} : \mathit{IDRange}]
\end{aligned}
$$

$$\mathit{NotActivity} \triangleq [\mathit{aid} \mapsto -1,\ b \mapsto \mathit{NotBlockID},\ \mathit{fid} \mapsto \mathit{NotID}]$$

Input Program: *Block* error used to simulate exceptions

$$
\begin{aligned}
\mathit{Block} \triangleq\ [&b : \mathit{BlockID} \cup \{\mathit{NotBlockID}\},\\
&\mathit{type} : \{\text{“NA”}, \text{“async”}, \text{“expr”}, \text{“finish”}, \text{“error”}, \text{“kill”}\},\\
&\mathit{dst} : \mathit{PLACE} \cup \{\mathit{NotPlace}\},\\
&\mathit{mxstmt} : \mathit{Nat},\\
&\mathit{stmts} : [\mathit{StmtID} \rightarrow \mathit{BlockID} \cup \{\mathit{EMPTY\_BLOCK},\ \mathit{NotBlockID}\}],\\
&\mathit{ran} : \text{BOOLEAN}\ ]
\end{aligned}
$$

$$\mathit{PlaceThread} \triangleq [\mathit{here} : \mathit{PLACE},\ \mathit{tid} : \mathit{ThreadID}]$$

$$\mathit{NotPlaceThread} \triangleq [\ \mathit{here} \mapsto \mathit{NotPlace},\ \ \mathit{tid} \mapsto \mathit{NotThreadID}]$$

$$
\begin{aligned}
\mathit{MasterStatus} \triangleq\ [&\mathit{status} : \{\text{“running”}, \text{“seekAdoption”}, \text{“convertDead”}\},\\
&\mathit{lastKilled} : \mathit{PLACE} \cup \{\mathit{NotPlace}\}]
\end{aligned}
$$

Finish Types

$$
\begin{aligned}
\mathit{FinishState} \triangleq\ [&\mathit{id} : \mathit{IDRange} \cup \{\mathit{NotID}\},\\
&\mathit{status} : \{\text{“unused”}, \text{“waiting”}, \text{“pendingRelease”}, \text{“forgotten”}\},\\
&\mathit{type} : \{\text{“distroot”}, \text{“distremote”}, \text{“NA”}\},\\
&\mathit{count} : \mathit{Nat},\\
&\mathit{here} : \mathit{PLACE} \cup \{\mathit{NotPlace}\},\\
&\mathit{parent} : \mathit{PIDRange} \cup \{\mathit{NotID}\}, \quad \text{used only in RESILIENT mode}\\
&\mathit{root} : \mathit{PIDRange} \cup \{\mathit{NotID}\}, \quad \text{root is the same as } \mathit{id} \text{ for root finishes}\\
&\mathit{isGlobal} : \text{BOOLEAN}\ , \quad \text{used by } \mathit{P0Finish}\\
&\mathit{eroot} : \mathit{PIDRange} \cup \{\mathit{NotID}\} \quad \text{root of the enclosing finish (used in } \mathit{PPoPP}14 \text{ dist finish)}\\
]&
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{ResilientFinishState} \triangleq [\\
&\quad \mathit{id} : \mathit{IDRange} \cup \{\mathit{NotID}\},\\
&\quad \mathit{parent} : \mathit{PIDRange} \cup \{\mathit{NotID}\},\\
&\quad \mathit{gfsRoot} : \mathit{PIDRange} \cup \{\mathit{NotID}\},\\
&\quad \mathit{gfsRootPlace} : \mathit{PLACE} \cup \{\mathit{NotPlace}\},
\end{aligned}
$$

$\quad numActive : Nat,$
$\quad live : [PLACE \rightarrow Nat],$
$\quad transit : [PLACE \rightarrow [PLACE \rightarrow Nat]],$
$\quad liveAdopted : [PLACE \rightarrow Nat],$
$\quad transitAdopted : [PLACE \rightarrow [PLACE \rightarrow Nat]],$
$\quad adopterId : IDRange \cup \{NotID\},$
$\quad isReleased : \text{BOOLEAN}$
$]$

$MasterFinish \triangleq [$
$\quad id : IDRange \cup \{NotID\},$
$\quad numActive : Nat,$
$\quad live : [PLACE \rightarrow Nat],$
$\quad transit : [PLACE \rightarrow [PLACE \rightarrow Nat]],$
$\quad liveAdopted : [PLACE \rightarrow Nat],$
$\quad transitAdopted : [PLACE \rightarrow [PLACE \rightarrow Nat]],$
$\quad children : \text{SUBSET } IDRange,$
$\quad backupPlace : PLACE \cup \{NotPlace\},$
$\quad isReleased : \text{BOOLEAN}$
$]$

$BackupFinish \triangleq [$
$\quad id : IDRange \cup \{NotID\},$
$\quad live : [PLACE \rightarrow Nat],$
$\quad transit : [PLACE \rightarrow [PLACE \rightarrow Nat]],$
$\quad children : \text{SUBSET } IDRange,$
$\quad isAdopted : \text{BOOLEAN} \ ,$
$\quad adoptedRoot : IDRange \cup \{NotID\},$
$\quad numActive : Nat,$
$\quad isReleased : \text{BOOLEAN}$
$]$

Message Types and Utilities

$NotMessage \triangleq [fid \mapsto NotID, src \mapsto NotPlace]$

$RemoteAsyncMessages \triangleq [mid : Nat,$
$\quad src : PLACE,$
$\quad dst : PLACE,$
$\quad type : \{\text{“async”}\},$
$\quad b : BlockID,$
$\quad fid : IDRange]$

$ReleaseFinishMessages \triangleq [mid : 0 \mathrel{..} 100,$
$\quad src : PLACE,$
$\quad dst : PLACE,$

$$\begin{array}{l} \mathit{fid} : \mathit{IDRange}, \\ \mathit{type} : \{\textsf{``releaseFinish''}\}] \end{array}$$

$$\mathit{AddChildMessages} \triangleq [\begin{array}{ll} \mathit{mid} & \mapsto \mathit{Nat}, \\ \mathit{src} & \mapsto \mathit{PLACE}, \\ \mathit{dst} & \mapsto \mathit{PLACE}, \\ \mathit{eroot} & \mapsto \mathit{IDRange}, \\ \mathit{fid} & \mapsto \mathit{IDRange}, \\ \mathit{type} & \mapsto \textsf{``addChild''}] \end{array}$$

$$\mathit{MasterTransitMessages} \triangleq [\begin{array}{ll} \mathit{mid} & \mapsto \mathit{Nat}, \\ \mathit{src} & \mapsto \mathit{PLACE}, \\ \mathit{dst} & \mapsto \mathit{PLACE}, \\ \mathit{target} & \mapsto \mathit{PLACE}, \\ \mathit{fid} & \mapsto \mathit{IDRange}, \\ \mathit{type} & \mapsto \{\textsf{``masterTransit''}, \textsf{``adopterTransit''}\}] \end{array}$$

$$\mathit{MasterLiveMessages} \triangleq [\begin{array}{ll} \mathit{mid} & \mapsto \mathit{Nat}, \\ \mathit{src} & \mapsto \mathit{PLACE}, \\ \mathit{source} & \mapsto \mathit{PLACE}, \\ \mathit{target} & \mapsto \mathit{PLACE}, \\ \mathit{dst} & \mapsto \mathit{PLACE}, \\ \mathit{fid} & \mapsto \mathit{IDRange}, \\ \mathit{aid} & \mapsto \mathit{Nat}, \\ \mathit{type} & \mapsto \{\textsf{``masterLive''}, \textsf{``adopterLive''}\}] \end{array}$$

$$\mathit{MasterCompletedMessages} \triangleq [\begin{array}{ll} \mathit{mid} & \mapsto \mathit{Nat}, \\ \mathit{src} & \mapsto \mathit{PLACE}, \\ \mathit{dst} & \mapsto \mathit{PLACE}, \\ \mathit{target} & \mapsto \mathit{PLACE}, \\ \mathit{fid} & \mapsto \mathit{IDRange}, \\ \mathit{finishEnd} & \mapsto \text{BOOLEAN}, \\ \mathit{type} & \mapsto \{\textsf{``masterCompleted''}, \textsf{``adopterCompleted''}\}] \end{array}$$

$$\mathit{BackupAddChild} \triangleq [\begin{array}{ll} \mathit{mid} & \mapsto \mathit{Nat}, \\ \mathit{src} & \mapsto \mathit{PLACE}, \\ \mathit{dst} & \mapsto \mathit{PLACE}, \\ \mathit{eroot} & \mapsto \mathit{IDRange}, \\ \mathit{fid} & \mapsto \mathit{IDRange}, \\ \mathit{type} & \mapsto \textsf{``backupAddChild''}] \end{array}$$

$$\mathit{AddChildDone} \triangleq [\begin{array}{ll} \mathit{mid} & \mapsto \mathit{Nat}, \\ \mathit{src} & \mapsto \mathit{PLACE}, \\ \mathit{dst} & \mapsto \mathit{PLACE}, \\ \mathit{eroot} & \mapsto \mathit{IDRange}, \\ \mathit{fid} & \mapsto \mathit{IDRange}, \\ \mathit{type} & \mapsto \{\textsf{``addChildDone''}, \textsf{``backupAddChildDone''}\}, \end{array}$$

$$\mathit{success} \mapsto \textsc{boolean}\ ]$$

$$\begin{array}{rl}
\mathit{BackupGetAdopter} \triangleq [& \mathit{mid} \mapsto \mathit{Nat}, \\
& \mathit{src} \mapsto \mathit{PLACE}, \\
& \mathit{dst} \mapsto \mathit{PLACE}, \\
& \mathit{fid} \mapsto \mathit{IDRange}, \\
& \mathit{actionType} \mapsto \{\text{“transit”}, \text{“completed”}, \text{“live”}\}, \\
& \mathit{aid} \mapsto \mathit{Nat}, \\
& \mathit{finishEnd} \mapsto \textsc{boolean}\ , \\
& \mathit{type} \mapsto \text{“backupGetAdopter”}]
\end{array}$$

$$\begin{array}{rl}
\mathit{GetAdopterDone} \triangleq [& \mathit{mid} \mapsto \mathit{Nat}, \\
& \mathit{src} \mapsto \mathit{PLACE}, \\
& \mathit{dst} \mapsto \mathit{PLACE}, \\
& \mathit{source} \mapsto \mathit{PLACE}, \\
& \mathit{target} \mapsto \mathit{PLACE}, \\
& \mathit{fid} \mapsto \mathit{IDRange}, \\
& \mathit{adoptedRoot} \mapsto \mathit{IDRange}, \\
& \mathit{actionType} \mapsto \{\text{“transit”}, \text{“completed”}, \text{“live”}\}, \\
& \mathit{aid} \mapsto \mathit{Nat}, \\
& \mathit{finishEnd} \mapsto \textsc{boolean}\ , \\
& \mathit{type} \mapsto \text{“backupGetAdopterDone”}]
\end{array}$$

$$\begin{array}{rl}
\mathit{MasterTransitDone} \triangleq [& \mathit{mid} \mapsto \mathit{Nat}, \\
& \mathit{src} \mapsto \mathit{PLACE}, \\
& \mathit{dst} \mapsto \mathit{PLACE}, \\
& \mathit{target} \mapsto \mathit{PLACE}, \\
& \mathit{fid} \mapsto \mathit{IDRange}, \\
& \mathit{type} \mapsto \text{“masterTransitDone”}, \\
& \mathit{isAdopter} \mapsto \textsc{boolean}\ , \\
& \mathit{submit} \mapsto \textsc{boolean}\ , \\
& \mathit{success} \mapsto \textsc{boolean}\ , \\
& \mathit{backupPlace} \mapsto \mathit{PLACE}]
\end{array}$$

$$\begin{array}{rl}
\mathit{BackupTransit} \triangleq [& \mathit{mid} \mapsto \mathit{Nat}, \\
& \mathit{src} \mapsto \mathit{PLACE}, \\
& \mathit{dst} \mapsto \mathit{PLACE}, \\
& \mathit{target} \mapsto \mathit{PLACE}, \\
& \mathit{fid} \mapsto \mathit{IDRange}, \\
& \mathit{isAdopter} \mapsto \textsc{boolean}\ , \\
& \mathit{adoptedFID} \mapsto \mathit{IDRange} \cup \{\mathit{NotID}\}, \\
& \mathit{type} \mapsto \text{“backupTransit”}]
\end{array}$$

$$\begin{array}{rl}
\mathit{BackupTransitDone} \triangleq [& \mathit{mid} \mapsto \mathit{Nat}, \\
& \mathit{src} \mapsto \mathit{PLACE},
\end{array}$$

$$\begin{array}{rl}
\mathit{dst} & \mapsto \mathit{PLACE}, \\
\mathit{target} & \mapsto \mathit{PLACE}, \\
\mathit{fid} & \mapsto \mathit{IDRange}, \\
\mathit{type} & \mapsto \text{“}\textsf{backupTransitDone}\text{”}, \\
\mathit{success} & \mapsto \textsc{Boolean}\ , \\
\mathit{isAdopter} & \mapsto \textsc{Boolean}\ , \\
\mathit{adoptedFID} & \mapsto \mathit{IDRange} \cup \{\mathit{NotID}\}]
\end{array}$$

$$\mathit{BackupLive} \triangleq [\begin{array}{rl}
\mathit{mid} & \mapsto \mathit{Nat}, \\
\mathit{src} & \mapsto \mathit{PLACE}, \\
\mathit{dst} & \mapsto \mathit{PLACE}, \\
\mathit{source} & \mapsto \mathit{PLACE}, \\
\mathit{target} & \mapsto \mathit{PLACE}, \\
\mathit{fid} & \mapsto \mathit{IDRange}, \\
\mathit{aid} & \mapsto \mathit{Nat}, \\
\mathit{type} & \mapsto \text{“}\textsf{backupLive}\text{”}, \\
\mathit{isAdopter} & \mapsto \textsc{Boolean}\ , \\
\mathit{adoptedFID} & \mapsto \mathit{IDRange} \cup \{\mathit{NotID}\}]
\end{array}$$

$$\mathit{BackupLiveDone} \triangleq [\begin{array}{rl}
\mathit{mid} & \mapsto \mathit{Nat}, \\
\mathit{src} & \mapsto \mathit{PLACE}, \\
\mathit{dst} & \mapsto \mathit{PLACE}, \\
\mathit{target} & \mapsto \mathit{PLACE}, \\
\mathit{source} & \mapsto \mathit{PLACE}, \\
\mathit{fid} & \mapsto \mathit{IDRange}, \\
\mathit{aid} & \mapsto \mathit{Nat}, \\
\mathit{type} & \mapsto \text{“}\textsf{backupLiveDone}\text{”}, \\
\mathit{success} & \mapsto \textsc{Boolean}\ , \\
\mathit{isAdopter} & \mapsto \textsc{Boolean}\ , \\
\mathit{adoptedFID} & \mapsto \mathit{IDRange} \cup \{\mathit{NotID}\}]
\end{array}$$

$$\mathit{MasterLiveDone} \triangleq [\begin{array}{rl}
\mathit{mid} & \mapsto \mathit{Nat}, \\
\mathit{src} & \mapsto \mathit{PLACE}, \\
\mathit{dst} & \mapsto \mathit{PLACE}, \\
\mathit{target} & \mapsto \mathit{PLACE}, \\
\mathit{source} & \mapsto \mathit{PLACE}, \\
\mathit{fid} & \mapsto \mathit{IDRange}, \\
\mathit{aid} & \mapsto \mathit{Nat}, \\
\mathit{type} & \mapsto \text{“}\textsf{masterLiveDone}\text{”}, \\
\mathit{submit} & \mapsto \textsc{Boolean}\ , \\
\mathit{success} & \mapsto \textsc{Boolean}\ , \\
\mathit{isAdopter} & \mapsto \textsc{Boolean}\ , \\
\mathit{backupPlace} & \mapsto \mathit{PLACE}]
\end{array}$$

$$
\begin{array}{lrl}
\mathit{BackupCompleted} \triangleq & [\mathit{mid} & \mapsto \mathit{Nat}, \\
& \mathit{src} & \mapsto \mathit{PLACE}, \\
& \mathit{dst} & \mapsto \mathit{PLACE}, \\
& \mathit{target} & \mapsto \mathit{PLACE}, \\
& \mathit{fid} & \mapsto \mathit{IDRange}, \\
& \mathit{type} & \mapsto \text{“backupCompleted”}, \\
& \mathit{isAdopter} & \mapsto \textsc{boolean}\ , \\
& \mathit{finishEnd} & \mapsto \textsc{boolean}\ ]
\end{array}
$$

$$
\begin{array}{lrl}
\mathit{MasterCompletedDone} \triangleq & [\mathit{mid} & \mapsto \mathit{Nat}, \\
& \mathit{src} & \mapsto \mathit{PLACE}, \\
& \mathit{dst} & \mapsto \mathit{PLACE}, \\
& \mathit{target} & \mapsto \mathit{PLACE}, \\
& \mathit{fid} & \mapsto \mathit{IDRange}, \\
& \mathit{type} & \mapsto \text{“masterCompletedDone”}, \\
& \mathit{success} & \mapsto \textsc{boolean}\ , \\
& \mathit{isAdopter} & \mapsto \textsc{boolean}\ , \\
& \mathit{backupPlace} & \mapsto \mathit{PLACE}]
\end{array}
$$

$$
\begin{array}{lrl}
\mathit{BackupCompletedDone} \triangleq & [\mathit{mid} & \mapsto \mathit{Nat}, \\
& \mathit{src} & \mapsto \mathit{PLACE}, \\
& \mathit{dst} & \mapsto \mathit{PLACE}, \\
& \mathit{target} & \mapsto \mathit{PLACE}, \\
& \mathit{fid} & \mapsto \mathit{IDRange}, \\
& \mathit{type} & \mapsto \text{“backupCompletedDone”}, \\
& \mathit{isAdopter} & \mapsto \textsc{boolean}\ , \\
& \mathit{success} & \mapsto \textsc{boolean}\ ]
\end{array}
$$

$$
\begin{array}{ll}
\mathit{DistFinishMessages} \triangleq & \mathit{AddChildMessages} \\
& \cup\ \mathit{MasterTransitMessages} \\
& \cup\ \mathit{MasterLiveMessages} \\
& \cup\ \mathit{MasterCompletedMessages} \\
& \cup\ \mathit{BackupAddChild} \\
& \cup\ \mathit{AddChildDone} \\
& \cup\ \mathit{BackupGetAdopter} \\
& \cup\ \mathit{GetAdopterDone} \\
& \cup\ \mathit{BackupTransit} \\
& \cup\ \mathit{MasterTransitDone} \\
& \cup\ \mathit{BackupTransitDone} \\
& \cup\ \mathit{BackupLive} \\
& \cup\ \mathit{MasterLiveDone} \\
& \cup\ \mathit{BackupLiveDone} \\
& \cup\ \mathit{BackupCompleted} \\
& \cup\ \mathit{MasterCompletedDone} \\
& \cup\ \mathit{BackupCompletedDone} \\
& \cup\ \mathit{ReleaseFinishMessages}
\end{array}
$$

```
$Messages \triangleq \quad RemoteAsyncMessages$
                 $\cup\ DistFinishMessages$

$BackupMessages \triangleq BackupAddChild$
                       $\cup\ BackupTransit$
                       $\cup\ BackupLive$
                       $\cup\ BackupCompleted$
                       $\cup\ BackupGetAdopter$

$SendMsg(m) \triangleq$
    $msgs' = msgs \cup \{m\}$

$RecvMsg(m) \triangleq$
    $msgs' = msgs \setminus \{m\}$

$ReplaceMsg(toRemove, toAdd) \triangleq$
    $msgs' = \ (msgs \setminus \{toRemove\}) \cup \{toAdd\}$

$ReplaceMsgSet(toRemove, toAddSet) \triangleq$
    $msgs' = \ (msgs \setminus \{toRemove\}) \cup toAddSet$
```

Predicates to extract the finish *id* from messages and *fstates*

```
$ExtractFIDFromMSG(src, dst, type) \triangleq$
    LET $mset \triangleq \{m \in msgs : \land\ m.src = src$
                                $\land\ m.dst \ \ = dst$
                                $\land\ m.type = type$
                                $\land\ m.fid \in IDRange$
                 $\}$
    IN   IF $mset = \{\}$ THEN $NotID$
         ELSE $($CHOOSE $x \in mset :$ TRUE$).fid$

$FindIncomingMSG(here, type) \triangleq$
    LET $mset \triangleq \{m \in msgs : \land\ m.dst \ \ = here$
                                $\land\ m.type = type$
                                $\land\ m.dst \notin killed$
                 $\}$
    IN   IF $mset = \{\}$ THEN $NotMessage$
         ELSE CHOOSE $x \in mset :$ TRUE

$FindMSG(type) \triangleq$
    LET $mset \triangleq \{m \in msgs : \land\ m.type = type$
                                $\land\ m.dst \notin killed$
                 $\}$
    IN   IF $mset = \{\}$ THEN $NotMessage$
         ELSE CHOOSE $x \in mset :$ TRUE
```

```
$GetActiveFID(type, here, pid) \triangleq$
    LET $mset \triangleq \{id \in IDRange : \land fstates[id].here = here$
                                          $\land fstates[id].root = pid$
                                          $\land fstates[id].type = type$
                                          $\land fstates[id].status =$ "waiting"
                    $\}$
    IN  IF $mset = \{\}$ THEN $NotID$
          ELSE (CHOOSE $x \in mset$ : TRUE)

$GetFinishHome(fid) \triangleq$
  IF $fid = NoParent$ THEN $PROG\_HOME$ ELSE $fstates[fid].here$

$GetEnclosingRoot(parent, me) \triangleq$
  IF $parent = NoParent$ THEN $NoParent$ ELSE $fstates[parent].root$
```

Predicate to extract thread ids with a specific status

```
$FindThread(here, status) \triangleq$
    LET $tset \triangleq \{t \in ThreadID : thrds[here][t].status = status\}$
    IN  IF $tset = \{\}$ THEN $NotThreadID$
          ELSE CHOOSE $x \in tset$ : TRUE

$FindThread2(here, statusSet) \triangleq$
    LET $tset \triangleq \{t \in ThreadID : thrds[here][t].status \in statusSet\}$
    IN  IF $tset = \{\}$ THEN $NotThreadID$
          ELSE CHOOSE $x \in tset$ : TRUE
```

Resilient Store Types and Utilities

```
$Adopter \triangleq [here : PLACE, child : IDRange \cup \{NotID\}, adopter : IDRange \cup \{NotID\}]$

$NotAdopter \triangleq [here \mapsto NotPlace, child \mapsto NotID, adopter \mapsto NotID]$

$ConvTask \triangleq [here : PLACE, fid : IDRange \cup \{NotID\}, pl : PLACE \cup \{NotPlace\}]$

$NotConvTask \triangleq [here \mapsto NotPlace, fid \mapsto NotID, pl \mapsto NotPlace]$

$GetBackup(p) \triangleq BACKUP[p]$
```

Utilities to increment sequences used to give unique ids to finish (*fseq*) messages (*mseq*), and activities (*aseq*)

```
$IncrFSEQ \triangleq$
  $seq' = [aseq \mapsto seq.aseq, fseq \mapsto seq.fseq + 1, mseq \mapsto seq.mseq]$

$IncrMSEQ(c) \triangleq$
  $seq' = [aseq \mapsto seq.aseq, fseq \mapsto seq.fseq, mseq \mapsto seq.mseq + c]$

$IncrASEQ \triangleq$
```

$seq' = [aseq \mapsto seq.aseq + 1,\ fseq \mapsto seq.fseq,\ mseq \mapsto seq.mseq]$

$IncrAll\ \triangleq$

$seq' = [aseq \mapsto seq.aseq + 1,\ fseq \mapsto seq.fseq + 1,\ mseq \mapsto seq.mseq + 1]$

* Modification History
* Last modified *Mon Dec* 11 20:59:18 *AEDT* 2017 by *u5482878*
* Last modified Sun *Dec* 10 16:09:57 *AEDT* 2017 by *shamouda*
* Created *Wed Sep* 27 09:26:18 *AEST* 2017 by *u5482878*