─── MODULE *Chameneos* ───

A speci cation of a 'concurrency game' requiring concurrent and symmetrical cooperation - https://*cedric.cnam.fr/* chiers/*RC474.pdf*

EXTENDS *Integers*

RECURSIVE $Sum(\_, \_)$
$Sum(f, S) \triangleq$ IF $S = \{\}$ THEN 0
　　　　　　　　　ELSE LET $x \triangleq$ CHOOSE $x \in S :$ TRUE
　　　　　　　　　　　 IN $f[x] + Sum(f, S \setminus \{x\})$

───

$Color \triangleq \{$"blue", "red", "yellow"$\}$
$Faded \triangleq$ CHOOSE $c : c \notin Color$

$Complement(c1, c2) \triangleq$ IF $c1 = c2$
　　　　　　　　　　　THEN $c1$
　　　　　　　　　　　ELSE CHOOSE $cid \in Color \setminus \{c1, c2\} :$ TRUE

───

N - number of total meeting after which *chameneoses* fade
M - number of *chameneoses*
CONSTANT N, M
ASSUME $N \in (Nat \setminus \{0\}) \wedge M \in (Nat \setminus \{0\})$

VARIABLE *chameneoses*, *meetingPlace*, *numMeetings*

$vars \triangleq \langle chameneoses, meetingPlace, numMeetings \rangle$

$ChameneosID \triangleq 1 .. M$
$MeetingPlaceEmpty \triangleq$ CHOOSE $e : e \notin ChameneosID$

$TypeOK \triangleq$
　　For each *chameneoses*, remember its current color and how many meetings it has been in.
　　$\wedge chameneoses \in [ChameneosID \rightarrow (Color \cup \{Faded\}) \times (0 .. N)]$
　　A *meetingPlace* (called Mall in the original paper) keeps track of the *chameneoses*
　　creature that is currently waiting to meet another creature.
　　$\wedge meetingPlace \in ChameneosID \cup \{MeetingPlaceEmpty\}$

$Init \triangleq \wedge chameneoses \in [ChameneosID \rightarrow Color \times \{0\}]$
　　　　$\wedge meetingPlace = MeetingPlaceEmpty$
　　　　$\wedge numMeetings = 0$

$Meet(cid) \triangleq$ IF $meetingPlace = MeetingPlaceEmpty$
　　　　　　THEN IF $numMeetings < N$
　　　　　　　　　　　chameneos enters empty meeting place
　　　　　　　　　THEN $\wedge meetingPlace' = cid$
　　　　　　　　　　　$\wedge$ UNCHANGED $\langle chameneoses, numMeetings \rangle$

$$\text{ELSE} \quad \wedge \; chameneoses' = [chameneoses \;\text{EXCEPT}\; ![cid] = \langle Faded, @[2]\rangle]$$
$$\wedge \;\text{UNCHANGED}\; \langle meetingPlace, numMeetings \rangle$$

$$\text{ELSE} \quad \wedge \; meetingPlace \neq cid$$
$$\wedge \; meetingPlace' = MeetingPlaceEmpty$$
$$\wedge \; chameneoses' =$$
$$\text{LET}\; newColor \;\triangleq\; Complement(chameneoses[cid][1],$$
$$chameneoses[meetingPlace][1])$$
$$\text{IN}\quad [chameneoses \;\text{EXCEPT}\; ![cid] = \langle newColor, @[2] + 1\rangle,$$
$$![meetingPlace] = \langle newColor, @[2] + 1\rangle]$$
$$\wedge \; numMeetings' = numMeetings + 1$$

Repeatedly try to enter meeting place for *chameneoses* that are not faded yet.
The system terminates once the color of all *chameneoses* is faded.

$$Next \;\triangleq\; \wedge \exists\, c \in \{x \in ChameneosID : chameneoses[x][1] \neq Faded\} : Meet(c)$$

$$Spec \;\triangleq\; Init \wedge \Box[Next]_{vars}$$

Upon termination, the sum of the (individual) meetings that all creates have
been in, is equal to $2 * N$. It is *not* guaranteed that all *chameneoses* have
been in a meeting with another *chameneoses*. See section A. Game termination
on page 5 of the original papaer).

$$SumMet \;\triangleq\; numMeetings = N \Rightarrow \text{LET}\; f[c \in ChameneosID] \;\triangleq\; chameneoses[c][2]$$
$$\text{IN}\quad Sum(f, ChameneosID) = 2 * N$$

$$\text{THEOREM}\; Spec \Rightarrow \Box SumMet$$