MODULE *fenestrate*

EXTENDS *Naturals*, *FiniteSets*, *TLC*

CONSTANTS
  DAYS, here specified by integers
  TIMES, here specified by integers as well
  MIN_WINDOWS,
  MIN_EXCLUSIONS

Selectors are functions which map each day to a boolean value representing
whether or not a day matches a window

$SELECTORS \triangleq [DAYS \rightarrow \text{BOOLEAN}]$

This is the set of all possible windows in tuple form, where each tuple
represents
- A selector, which is whether or not this window is active on a given *day*
– A start *time*
– An end time. If the end time is less than the start time, this window spans the next day.

$WINDOWS \triangleq SELECTORS \times TIMES \times TIMES$

All possible datetimes

$DATETIMES \triangleq DAYS \times TIMES$

Whether or not the given *datetime* matches the window.

They match if the window's *from_time* $<$ *to_time* and the *datetime*, the *datetime* is between those, and the selector matches the *datetime*'s date

or

the window's *from_time* $>$ *to_time* the *datetime* $<$ *to_time* and the selector matches the day BEFORE the *datetime*'s date

```
in_window(window, datetime) ≜
  LET
    selector ≜ window[1]
    day ≜ datetime[1]
    time ≜ datetime[2]
    from ≜ window[2]
    to ≜ window[3]
  IN
    ∧ from ≤ to ⇒ ∧ from ≤ time
                  ∧ time ≤ to
                   ∧ selector[day]
    ∧ from > to ⇒ ∧ time < to
                  ∧ selector[day − 1]
```

this needs to check now against windows for the previous day.

```
in_nonexcluded_windows(windows, exclusions, now) ≜
    ∧ ∃ w ∈ windows : in_window(w, now)
```

$\land \neg \exists\, e \in exclusions : in\_window(e, now)$

```
--algorithm fenestrate

 test
variables
  $now \in DATETIMES$,
  $windows = \text{CHOOSE } w \in \text{SUBSET } WINDOWS : Cardinality(w) > MIN\_WINDOWS$,
  $exclusions = \text{CHOOSE } e \in \text{SUBSET } WINDOWS : Cardinality(e) > MIN\_EXCLUSIONS$,
  $result$ ;
begin
check_in_window:
  $result := in\_nonexcluded\_windows(windows, exclusions, now)$ ;
  print $\langle result, now \rangle$ ;
end algorithm
end algorithm;
```

BEGIN TRANSLATION ($chksum(pcal) =$ "8935074d" $\land\ chksum(tla) =$ "7c765564")

CONSTANT $defaultInitValue$

VARIABLES $now, windows, exclusions, result, pc$

$vars \triangleq \langle now, windows, exclusions, result, pc \rangle$

$Init \triangleq$ Global variables

$\land\ now \in DATETIMES$

$\land\ windows = (\text{CHOOSE } w \in \text{SUBSET } WINDOWS : Cardinality(w) > MIN\_WINDOWS)$

$\land\ exclusions = (\text{CHOOSE } e \in \text{SUBSET } WINDOWS : Cardinality(e) > MIN\_EXCLUSIONS)$

$\land\ result = defaultInitValue$

$\land\ pc =$ "check_in_window"

$check\_in\_window \triangleq \land\ pc =$ "check_in_window"

$\land\ result' = in\_nonexcluded\_windows(windows, exclusions, now)$

$\land\ PrintT(\langle result', now \rangle)$

$\land\ pc' =$ "Done"

$\land$ UNCHANGED $\langle now, windows, exclusions \rangle$

Allow infinite stuttering to prevent deadlock on termination.

$Terminating \triangleq pc =$ "Done" $\land$ UNCHANGED $vars$

$Next \triangleq check\_in\_window$

$\lor\ Terminating$

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

$Termination \triangleq \Diamond(pc =$ "Done"$)$

END TRANSLATION

\* Modification History
\* Last modified *Mon Jun* 21 09:11:40 *CDT* 2021 by *vputz*
\* Created *Wed Jun* 16 10:07:32 *CDT* 2021 by *vputz*