MODULE *Main*

EXTENDS *TLC*, *Sequences*, *FiniteSets*, *Integers*, *PT*
CONSTANTS *INGREDIENTS*, *MAX_QTTY*, *USERS*

Ingredient records have an type (banana...) and a integer quantity
AllIngredientSets $\triangleq$
$\{x \in (\text{SUBSET } [type : INGREDIENTS,\ qtty : 1 \,..\, MAX\_QTTY]) :$
$\neg\exists\, i, j \in x : i \neq j \wedge i.type = j.type\} \setminus \{\{\}\}$

--algorithm *recettes*
variables $fridge = \{\}$,
$recipe\_items = [u \in USERS \mapsto \{\}]$ **;**

define
Recipes with all ingredients available in the *fridge*
CraftableRecipes $\triangleq$
$\{r \in AllIngredientSets : \forall\, item \in r : \exists\, elt \in fridge :$
$item.type = elt.type \wedge item.qtty \leq elt.qtty\}$

EnoughInFridgeForARecipe $\triangleq$
$Cardinality(CraftableRecipes) > 0$

NoOtherUserInTheKitchen(*user*) $\triangleq$
$\neg\exists\, u \in USERS : u \neq user \wedge recipe\_items[u] \neq \{\}$

GetOneRecipe $\triangleq$
CHOOSE $r \in CraftableRecipes$: TRUE

GetBoughtItems $\triangleq$
CHOOSE $is \in AllIngredientSets$: TRUE
end define ;

fair + process *client* $\in$ *USERS*
variable $bought\_items = \{\}$,
$n\_recipes = 0$ **;**

begin *Client*:
while TRUE **do**
either
BuyIngredients:
$bought\_items := GetBoughtItems$ **;**
FillFridge:
while $bought\_items \neq \{\}$ **do**
with $b\_item \in bought\_items$ **do**
$fridge :=$ IF $\exists\, i \in fridge : i.type = b\_item.type$
THEN LET $f\_item \triangleq$ CHOOSE $i \in fridge : i.type = b\_item.type$
IN $(fridge \setminus \{f\_item\}) \cup \{[type \mapsto b\_item.type,\ qtty \mapsto f\_item.qtty + b\_ite$

```
                        ELSE  fridge ∪ {b_item} ;
                    bought_items := bought_items \ {b_item} ;
                end with ;
            end while ;
        or
            await EnoughInFridgeForARecipe ;
            ChooseRecipe:
            await EnoughInFridgeForARecipe ∧ NoOtherUserInTheKitchen(self) ;
            recipe_items[self] := GetOneRecipe ;
            MakeRecipe:
            while recipe_items[self] ≠ {} do
                with r_item ∈ recipe_items[self] do
                    fridge := LET f_item ≜ CHOOSE i ∈ fridge : i.type = r_item.type
                                  new_item ≜ [type ↦ r_item.type, qtty ↦ f_item.qtty − r_item.qtty]
                              IN  (fridge \ {f_item}) ∪ IF new_item.qtty > 0 THEN {new_item} ELSE {} ;
                    recipe_items[self] := recipe_items[self] \ {r_item} ;
                end with ;
            end while ;
            n_recipes := n_recipes + 1 ;
        end either ;
    end while ;
end process ;
end algorithm ;
```

BEGIN TRANSLATION ($chksum(pcal) =$ "561b9c1c" $\wedge\ chksum(tla) =$ "f186b71f")

VARIABLES *pc*, *fridge*, *recipe_items*

define statement

$$CraftableRecipes \triangleq \{r \in AllIngredientSets : \forall\, item \in r : \exists\, elt \in fridge : item.type = elt.type \wedge item.qtty \leq elt.qtty\}$$

$$EnoughInFridgeForARecipe \triangleq Cardinality(CraftableRecipes) > 0$$

$$NoOtherUserInTheKitchen(user) \triangleq \neg\exists\, u \in USERS : u \neq user \wedge recipe\_items[u] \neq \{\}$$

$$GetOneRecipe \triangleq \text{CHOOSE } r \in CraftableRecipes : \text{TRUE}$$

$$GetBoughtItems \triangleq \text{CHOOSE } is \in AllIngredientSets : \text{TRUE}$$

VARIABLES *bought_items*, *n_recipes*

$$vars \triangleq \langle pc, fridge, recipe\_items, bought\_items, n\_recipes\rangle$$

$ProcSet \triangleq (USERS)$

$Init \triangleq$ Global variables
$\land fridge = \{\}$
$\land recipe\_items = [u \in USERS \mapsto \{\}]$
Process client
$\land bought\_items = [self \in USERS \mapsto \{\}]$
$\land n\_recipes = [self \in USERS \mapsto 0]$
$\land pc = [self \in ProcSet \mapsto \text{"Client"}]$

$Client(self) \triangleq \land pc[self] = \text{"Client"}$
$\land \lor \land pc' = [pc \text{ EXCEPT } ![self] = \text{"BuyIngredients"}]$
$\lor \land EnoughInFridgeForARecipe$
$\land pc' = [pc \text{ EXCEPT } ![self] = \text{"ChooseRecipe"}]$
$\land \text{UNCHANGED } \langle fridge, recipe\_items, bought\_items, n\_recipes \rangle$

$BuyIngredients(self) \triangleq \land pc[self] = \text{"BuyIngredients"}$
$\land bought\_items' = [bought\_items \text{ EXCEPT } ![self] = GetBoughtItems]$
$\land pc' = [pc \text{ EXCEPT } ![self] = \text{"FillFridge"}]$
$\land \text{UNCHANGED } \langle fridge, recipe\_items, n\_recipes \rangle$

$FillFridge(self) \triangleq \land pc[self] = \text{"FillFridge"}$
$\land \text{IF } bought\_items[self] \neq \{\}$
$\text{THEN } \land \exists\, b\_item \in bought\_items[self] :$
$\land fridge' = (\text{IF } \exists\, i \in fridge : i.type = b\_item.type$
$\text{THEN LET } f\_item \triangleq \text{CHOOSE } i \in fridge : i.type = b\_item.t$
$\text{IN } (fridge \setminus \{f\_item\}) \cup \{[type \mapsto b\_item.type,\ qtty \mapsto$
$\text{ELSE } fridge \cup \{b\_item\})$
$\land bought\_items' = [bought\_items \text{ EXCEPT } ![self] = bought\_items[self] \setminus \{$
$\land pc' = [pc \text{ EXCEPT } ![self] = \text{"FillFridge"}]$
$\text{ELSE } \land pc' = [pc \text{ EXCEPT } ![self] = \text{"Client"}]$
$\land \text{UNCHANGED } \langle fridge, bought\_items \rangle$
$\land \text{UNCHANGED } \langle recipe\_items, n\_recipes \rangle$

$ChooseRecipe(self) \triangleq \land pc[self] = \text{"ChooseRecipe"}$
$\land EnoughInFridgeForARecipe \land NoOtherUserInTheKitchen(self)$
$\land recipe\_items' = [recipe\_items \text{ EXCEPT } ![self] = GetOneRecipe]$
$\land pc' = [pc \text{ EXCEPT } ![self] = \text{"MakeRecipe"}]$
$\land \text{UNCHANGED } \langle fridge, bought\_items, n\_recipes \rangle$

$MakeRecipe(self) \triangleq \land pc[self] = \text{"MakeRecipe"}$
$\land \text{IF } recipe\_items[self] \neq \{\}$
$\text{THEN } \land \exists\, r\_item \in recipe\_items[self] :$
$\land fridge' = (\text{LET } f\_item \triangleq \text{CHOOSE } i \in fridge : i.type = r\_item.type$
$new\_item \triangleq [type \mapsto r\_item.type,\ qtty \mapsto f\_item.qtty$
$\text{IN } (fridge \setminus \{f\_item\}) \cup \text{IF } new\_item.qtty > 0 \text{ THEN } \{new$
$\land recipe\_items' = [recipe\_items \text{ EXCEPT } ![self] = recipe\_items[self] \setminus \{$

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“MakeRecipe”}] \\
\qquad\qquad\qquad\qquad\qquad\qquad \land \text{ UNCHANGED } n\_recipes \\
\qquad\qquad\qquad\qquad\quad \text{ELSE} \quad \land\ n\_recipes' = [n\_recipes \text{ EXCEPT } ![self] = n\_recipes[self] + 1] \\
\qquad\qquad\qquad\qquad\qquad\qquad \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“Client”}] \\
\qquad\qquad\qquad\qquad\qquad\qquad \land \text{ UNCHANGED } \langle fridge,\ recipe\_items \rangle \\
\qquad\qquad\qquad\qquad\qquad \land \text{ UNCHANGED } bought\_items
\end{array}
$$

$$
\begin{array}{l}
client(self) \triangleq\ Client(self) \lor BuyIngredients(self) \lor FillFridge(self) \\
\qquad\qquad\qquad\qquad\quad \lor\ ChooseRecipe(self) \lor MakeRecipe(self)
\end{array}
$$

$$
Next \triangleq\ (\exists\, self \in USERS : client(self))
$$

$$
\begin{array}{l}
Spec \triangleq\ \land\ Init \land \Box[Next]_{vars} \\
\qquad\qquad\ \land \forall\, self \in USERS : \mathrm{SF}_{vars}(client(self))
\end{array}
$$

END TRANSLATION

INVARIANTS

$$
\begin{array}{l}
TypeOK \triangleq \\
\quad \land \quad fridge = \{\} \lor \forall\, i \in fridge : i.type \in INGREDIENTS \land i.qtty > 0
\end{array}
$$

PROPERTIES

$$
BuysIngredientsInFridge \triangleq\ \exists\, self \in ProcSet : pc[self] = \text{“BuyIngredients”} \rightsquigarrow fridge \neq \{\}
$$

$$
MakeRecipeHappens \triangleq\ \Diamond(\exists\, self \in ProcSet : pc[self] = \text{“MakeRecipe”})
$$

* Modification History
* Last modified Sun *Jul* 21 00:13:07 *CEST* 2024 by *davd*33
* Created *Wed Jul* 17 21:03:29 *CEST* 2024 by *davd*33