MODULE *SimpleAllocator*

EXTENDS *FiniteSets*

CONSTANTS *Clients*, *Resources*

ASSUME $IsFiniteSet(Resources)$

VARIABLES

$\quad unsat,$ $\quad$ $unsat[c]$ denotes the outstanding requests of client c

$\quad alloc$ $\quad$ $alloc[c]$ denotes the resources allocated to client c

---

$TypeInvariant \triangleq$

$\quad \land\ unsat \in [Clients \rightarrow \text{SUBSET}\ Resources]$

$\quad \land\ alloc\ \ \in [Clients \rightarrow \text{SUBSET}\ Resources]$

$available \triangleq$ Set of resources free for allocation

$\quad Resources \setminus (\text{UNION}\ \{alloc[c] : c \in Clients\})$

$Init \triangleq$ Initially, no resources have been requested or allocated

$\quad \land\ unsat = [c \in Clients \mapsto \{\}]$

$\quad \land\ alloc\ \ = [c \in Clients \mapsto \{\}]$

$Request(c, S) \triangleq$ Client c requests set S of resources

$\quad \land\quad S \neq \{\} \land unsat[c] = \{\} \land alloc[c] = \{\}$

$\quad \land\quad unsat' = [unsat\ \text{EXCEPT}\ ![c] = S]$

$\quad \land\quad \text{UNCHANGED}\ alloc$

$Allocate(c, S) \triangleq$ Set S of available resources are allocated to client c

$\quad \land\ S \neq \{\} \land S \subseteq available \cap unsat[c]$

$\quad \land\ alloc'\ \ = [alloc\ \text{EXCEPT}\ ![c]\ \ = @ \cup S]$

$\quad \land\ unsat' = [unsat\ \text{EXCEPT}\ ![c] = @ \setminus S]$

$Return(c, S) \triangleq$ Client c returns a set of resources that it holds

$\quad \land\ S \neq \{\} \land S \subseteq alloc[c]$

$\quad \land\ alloc' = [alloc\ \text{EXCEPT}\ ![c] = @ \setminus S]$

$\quad \land\ \text{UNCHANGED}\ unsat$

$Next \triangleq$ The system's next-state relation

$\quad \exists\, c \in Clients, S \in \text{SUBSET}\ Resources :$

$\qquad Request(c, S) \lor Allocate(c, S) \lor Return(c, S)$

$vars \triangleq \langle unsat, alloc \rangle$

$SimpleAllocator \triangleq$ The complete high-level specification

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

$\quad \land\ \forall\, c \in Clients : \mathrm{WF}_{vars}(Return(c, alloc[c]))$

$\quad \land\ \forall\, c \in Clients : \mathrm{SF}_{vars}(\exists\, S \in \text{SUBSET}\ Resources : Allocate(c, S))$

$Safety \triangleq \forall\, c1, c2 \in Clients : c1 \neq c2 \Rightarrow alloc[c1] \cap alloc[c2] = \{\}$

$Liveness \triangleq \forall\, c \in Clients, r \in Resources : r \in unsat[c] \leadsto r \in alloc[c]$

\* Modification History

\* Last modified Sun May 28 20:16:09 *CST* 2017 by ics-ant

\ * Created Sun May 28 19:44:57 CST 2017 by ics-ant