MODULE *Harmony*

EXTENDS *Integers*, *Sequences*

VARIABLE *CTXBAG*, *SHARED*

some helper functions

add *var* with *val* to map

$NMap(var, val, map) \triangleq [x \in ((\text{DOMAIN } map) \cup \{var\}) \setminus \{\text{“FALSE”}\} \mapsto \text{IF } x = var \text{ THEN } val \text{ ELSE } map[x]]$

remove *var* from map, until empty map, *i.e.*, FALSE ↦ FALSE

$NMap2(var, map) \triangleq [x \in ((\text{DOMAIN } map) \setminus \{var\}) \cup \{\text{“FALSE”}\} \mapsto \text{IF } x \in \text{DOMAIN } map \text{ THEN } map[x] \text{ ELSE}$

remove *var* from map

$NMapReturn(var, map) \triangleq [x \in ((\text{DOMAIN } map) \setminus \{var\}) \mapsto map[x]]$

$\text{RECURSIVE } NTail(\_, \_)$

$\text{RECURSIVE } NHead(\_, \_)$

$\text{RECURSIVE } AddMult(\_, \_, \_)$

$AddMult(var\_tup, val\_tup, map) \triangleq \text{IF } Len(var\_tup) = 1 \text{ THEN } [x \in ((\text{DOMAIN } map) \cup \{Head(var\_tup)\}) \setminus \{$

$\qquad \text{ELSE } [x \in ((\text{DOMAIN } AddMult(Tail(var\_tup), Tail(val\_tup), map)) \cup$

the last *n* elements of the list

$NTail(n, tup) \triangleq \text{IF } n = 1 \text{ THEN } Tail(tup) \text{ ELSE } NTail(n - 1, Tail(tup))$

the first *n* elements of a *tup*

$NHead(n, tup) \triangleq \text{IF } n = 1 \text{ THEN } \langle Head(tup) \rangle \text{ ELSE } NHead(n - 1, Tail(tup)) \circ \langle Head(tup) \rangle$

nth element of a *tup*

$SpawnHead(ctx) \triangleq NHead(3, CTXBAG[ctx].stack)$

$SpawnTail(ctx) \triangleq NTail(3, CTXBAG[ctx].stack)$

empty record

$e\_rec \triangleq [\text{FALSE} \mapsto \text{FALSE}]$

a new context

$new\_ctx \triangleq [pc \mapsto 0, stack \mapsto \langle\langle\rangle\rangle, vars \mapsto e\_rec, active \mapsto \text{FALSE}, spn \mapsto \text{FALSE}]$

initial context is marked as spawned;
return checks if context is either in a “spawn state” or “applied state”

$init\_ctx \triangleq [pc \mapsto 0, stack \mapsto \langle\langle\rangle\rangle, vars \mapsto e\_rec, active \mapsto \text{TRUE}, spn \mapsto \text{TRUE}]$

Harmony Initial State

$HarmonyInit \triangleq$ global variable

$\quad \land SHARED = e\_rec$ start empty

$\quad \land CTXBAG = [c0 \mapsto init\_ctx, c1 \mapsto new\_ctx, c2 \mapsto new\_ctx]$

push *val* onto head of *ctx* stack

$Push(ctx, val, PC) \triangleq$

$\quad \land \ CTXBAG[ctx].pc = PC$

$\quad \land \ CTXBAG[ctx].active = \text{TRUE}$

$\quad \land \ CTXBAG' = [CTXBAG \text{ EXCEPT } ![ctx].pc = PC + 1, ![ctx].stack = \langle val \rangle \circ CTXBAG[ctx].stack]$

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

thread store

$\mathit{StoreVar}(ctx, var, PC) \triangleq$
$\land\ \mathit{CTXBAG}[ctx].pc = PC$
$\land\ \mathit{CTXBAG}[ctx].active = \text{TRUE}$
$\land\ \mathit{CTXBAG}' = [\mathit{CTXBAG}\ \text{EXCEPT}\ ![ctx].pc = PC + 1,\ ![ctx].stack = \mathit{Tail}(\mathit{CTXBAG}[ctx].stack),\ ![ctx].var$
$\land\ \text{UNCHANGED}\ \mathit{SHARED}$

shared store

$\mathit{Store}(ctx, var, PC) \triangleq$
$\land\ \mathit{CTXBAG}[ctx].pc = PC$
$\land\ \mathit{CTXBAG}[ctx].active = \text{TRUE}$
$\land\ \mathit{CTXBAG}' = [\mathit{CTXBAG}\ \text{EXCEPT}\ ![ctx].pc = PC + 1,\ ![ctx].stack = \mathit{Tail}(\mathit{CTXBAG}[ctx].stack)]$
$\land\ \mathit{SHARED}' = \mathit{NMap}(var, \mathit{Head}(\mathit{CTXBAG}[ctx].stack), \mathit{SHARED})$

$\mathit{Jump}(ctx, PC, PC\_new) \triangleq$
$\land\ \ \mathit{CTXBAG}[ctx].pc = PC$
$\land\ \ \mathit{CTXBAG}[ctx].active = \text{TRUE}$
$\land\ \ \mathit{CTXBAG}' = [\mathit{CTXBAG}\ \text{EXCEPT}\ ![ctx].pc = PC\_new]$
$\land\ \ \text{UNCHANGED}\ \mathit{SHARED}$

push the value of a shared variable onto the context stack

$\mathit{Load}(ctx, var\_name, PC) \triangleq$
$\land\ \ \mathit{CTXBAG}[ctx].pc = PC$
$\land\ \ \mathit{CTXBAG}[ctx].active = \text{TRUE}$

push the value of a shared variable onto the stack

$\land\ \mathit{CTXBAG}' = [\mathit{CTXBAG}\ \text{EXCEPT}\ ![ctx].pc = PC + 1,\ ![ctx].stack = \langle \mathit{SHARED}[var\_name] \rangle \circ \mathit{CTXBAG}[c$
$\land\ \text{UNCHANGED}\ \mathit{SHARED}$

push the value of a thread variable onto the stack

$\mathit{LoadVar}(ctx, var\_name, PC) \triangleq$
$\land\ \mathit{CTXBAG}[ctx].pc = PC$
$\land\ \mathit{CTXBAG}[ctx].active = \text{TRUE}$
$\land\ \mathit{CTXBAG}' = [\mathit{CTXBAG}\ \text{EXCEPT}\ ![ctx].pc = PC + 1,\ ![ctx].stack = \langle \mathit{CTXBAG}[ctx].vars[var\_name] \rangle \circ \mathit{CT}$
$\land\ \text{UNCHANGED}\ \mathit{SHARED}$

$\mathit{Spawn}(ctxa, PC) \triangleq$
$\land\ \mathit{CTXBAG}[ctxa].pc = PC$
$\land\ \mathit{CTXBAG}[ctxa].active = \text{TRUE}$
$\land\ \text{LET}\ \mathit{SpStk} \triangleq \mathit{SpawnHead}(ctxa) \text{IN}$
$\quad \text{LET}\ ctxb \triangleq \text{CHOOSE}\ x \in \text{DOMAIN}\ \mathit{CTXBAG} : \mathit{CTXBAG}[x].active = \text{FALSEIN}$
$\quad\ \land\ \mathit{CTXBAG}' = [\mathit{CTXBAG}\ \text{EXCEPT}\ ![ctxa].pc = PC + 1,\ ![ctxa].stack = \mathit{SpawnTail}(ctxa),\ ![ctxb].pc = \mathit{He}$
$\land\ \text{UNCHANGED}\ \mathit{SHARED}$

delete thread variable *var*

$\mathit{DelVar}(ctx, var, PC) \triangleq$
$\land\ \mathit{CTXBAG}[ctx].pc = PC$
$\land\ \mathit{CTXBAG}[ctx].active = \text{TRUE}$

```
  $\land\ CTXBAG' = [CTXBAG$ EXCEPT $![ctx].pc = PC + 1,\ ![ctx].vars = NMap2(var,\ CTXBAG[ctx].vars)]$
  $\land$ UNCHANGED $SHARED$

 take top of the context's stack and assign it to *Frame* instruction arguments *args*
 *TODO* want to do store *var* on possibly a tuple, only works for single *var* now
$Frame(ctx,\ args,\ PC)\ \triangleq$
  $\land\ CTXBAG[ctx].pc = PC$
  $\land\ CTXBAG[ctx].active = $ TRUE
  $\land\ CTXBAG[ctx].spn = $ TRUE
  $\land\ CTXBAG' = [CTXBAG$ EXCEPT $![ctx].pc = PC + 1,\ ![ctx].stack = Tail(CTXBAG[ctx].stack),\ ![ctx].var$
  $\land$ UNCHANGED $SHARED$

$Return(ctx,\ PC)\ \triangleq$
  $\land\ CTXBAG[ctx].pc = PC$
  $\land\ CTXBAG[ctx].active\ = $ TRUE
  $\land$ IF $CTXBAG[ctx].spn = $ TRUE THEN
      $\land\ CTXBAG' = [CTXBAG$ EXCEPT $![ctx].active = $ FALSE$]$
     ELSE
      $\land\ CTXBAG' = [CTXBAG$ EXCEPT $![ctx].pc = Head(CTXBAG[ctx].stack),\ ![ctx].stack = Tail(CTXBAG$
  $\land$ UNCHANGED $SHARED$
```

* Modification History
* Last modified *Mon Nov* 29 22:36:38 *EST* 2021 by *noah*
* Last modified *Thu Nov* 18 16:26:44 *EST* 2021 by *arielkellison*
* Created *Tue Nov* 02 18:59:20 *EDT* 2021 by *arielkellison*