```
- MODULE LAM_ISA_Move -
 1 [
2 Extends Naturals, Sequences
4 CONSTANTS InstrCount, RegisterCount, Nil
   RegisterKinds \stackrel{\Delta}{=} \{ \text{"local"}, \text{"global"} \}
       --algorithm LAM_ISA_Move
    variables
9
      qlobal\_registers = (0 .. RegisterCount) \times \{Nil\},
10
      local\_registers = (0 .. RegisterCount) \times \{Nil\},
11
      current\_count = InstrCount,
12
      current\_move = Nil
13
14
    define
16
      AllRegistersAreValid \stackrel{\triangle}{=} \lor current\_move = Nil
18
                                  \lor \land current\_move.dst[1] \in RegisterKinds
19
                                     \land current\_move.src[1] \in RegisterKinds
20
                                     \land current\_move.dst[2] \in (0 .. RegisterCount)
21
                                     \land current\_move.src[2] \in (0 .. RegisterCount)
22
    end define;
24
    procedure perform_move(move)begin
26
27
      if move.dst[1] = "global" \land move.src[1] = "global" then
28
        global\_registers[move.dst[2]] := global\_registers[move.src[2]]
29
       elsif move.dst[1] = "local" \land move.src[1] = "local" then
30
        local\_registers[move.dst[2]] := local\_registers[move.src[2]]
31
       elsif move.dst[1] = "global" \land move.src[1] = "local" then
32
        global\_registers[move.dst[2]] := local\_registers[move.src[2]]
33
       elsif move.dst[1] = "local" \land move.src[1] = "global" then
34
        local\_registers[move.dst[2]] := global\_registers[move.src[2]]
35
36
      end if;
      return;
37
    end procedure;
38
    begin
40
      Run:
41
        while current\_count > 0 do
42
          MakeMove:
43
            current\_move := [
44
45
                  CHOOSE k \in RegisterKinds:
46
                  CHOOSE n \in (0 ... RegisterCount) : \langle k, n \rangle,
47
                dst \mapsto
```

```
Choose k \in RegisterKinds:
49
                    CHOOSE n \in (0 ... RegisterCount) : \langle k, n \rangle
50
             ];
51
             current\_count := current\_count - 1;
52
           call perform_move(current_move);
53
        end while;
54
    end algorithm ;
55
     BEGIN TRANSLATION (chksum(pcal) = "c6cc4c7" \land chksum(tla) = "c697ae07")
    CONSTANT defaultInitValue
58
    VARIABLES global_registers, local_registers, current_count, current_move, pc,
59
60
     define statement
62
    AllRegistersAreValid \triangleq
                                 \lor current\_move = Nil
63
                                  \lor \land current\_move.dst[1] \in RegisterKinds
64
                                      \land current\_move.src[1] \in RegisterKinds
65
                                      \land current\_move.dst[2] \in (0 .. RegisterCount)
66
                                      \land current\_move.src[2] \in (0 ... RegisterCount)
67
    VARIABLE move
69
            \triangleq \langle global\_registers, local\_registers, current\_count, current\_move, pc,
71
                 stack, move \rangle
72
                 Global variables
    Init
74
                \land global\_registers = (0 .. RegisterCount) \times \{Nil\}
75
                \land local\_registers = (0 .. RegisterCount) \times \{Nil\}
76
                \land current\_count = InstrCount
77
                 \land current\_move = Nil
78
79
                 Procedure perform\_move
80
                \land move = defaultInitValue
                \wedge stack = \langle \rangle
81
                \wedge pc = "Run"
82
    Move \stackrel{\triangle}{=} \land pc = \text{``Move''}
84
                \land IF move.dst[1] = "global" <math>\land move.src[1] = "global"
85
                       THEN \land global\_registers' = [global\_registers \ EXCEPT \ ![move.dst[2]] = global\_registers[move.sr]
86
                               ↑ UNCHANGED local_registers
87
                       ELSE \land IF move.dst[1] = "local" <math>\land move.src[1] = "local"
88
                                      THEN \land local\_registers' = [local\_registers \ EXCEPT \ ![move.dst[2]] = local\_registers
89
                                              ∧ UNCHANGED global_registers
90
                                      ELSE \land IF move.dst[1] = "global" <math>\land move.src[1] = "local"
91
                                                     Then \land global\_registers' = [global\_registers \ \texttt{Except} \ ! [move.dst[2]] =
92
                                                             ∧ UNCHANGED local_registers
93
                                                     ELSE \land IF move.dst[1] = "local" <math>\land move.src[1] = "global"
94
                                                                    THEN \land local\_registers' = [local\_registers \ EXCEPT \ ![moverement]]
```

95

```
ELSE \land TRUE
 96
                                                                                    ∧ UNCHANGED local_registers
 97
                                                                   ∧ UNCHANGED global_registers
 98
                   \wedge pc' = Head(stack).pc
 99
                   \land move' = Head(stack).move
100
                   \wedge stack' = Tail(stack)
101
                   \land UNCHANGED \langle current\_count, current\_move \rangle
102
     perform\_move \triangleq Move
104
      Run \stackrel{\triangle}{=} \land pc = \text{``Run''}
106
                 \land if current\_count > 0
107
                        THEN \wedge pc' = "MakeMove"
108
                         ELSE \wedge pc' = "Done"
109
                 \land UNCHANGED \langle global\_registers, local\_registers, current\_count,
110
111
                                       current\_move, stack, move \rangle
      MakeMove \stackrel{\triangle}{=} \land pc = \text{``MakeMove''}
113
                          \land current\_move' =
114
                                                       src \mapsto
115
                                                         CHOOSE k \in RegisterKinds:
116
                                                         CHOOSE n \in (0 ... RegisterCount) : \langle k, n \rangle,
117
                                                       dst \mapsto
118
                                                         Choose k \in RegisterKinds:
119
                                                         CHOOSE n \in (0 ... RegisterCount) : \langle k, n \rangle
120
121
                          \land current\_count' = current\_count - 1
122
                          \land \land move' = current\_move'
123
                             \land stack' = \langle [procedure \mapsto "perform\_move",
124
                                                           \mapsto "Run",
125
                                                          \mapsto move \rangle
126
                                             move
                                             \circ \ stack
127
                          \wedge pc' = "Move"
128
                          \land UNCHANGED \langle global\_registers, local\_registers \rangle
129
       Allow infinite stuttering to prevent deadlock on termination.
131
      Terminating \stackrel{\triangle}{=} pc = "Done" \land UNCHANGED vars
132
     Next \triangleq perform\_move \lor Run \lor MakeMove
134
                      \vee Terminating
135
     Spec \triangleq Init \wedge \Box [Next]_{vars}
137
      Termination \stackrel{\triangle}{=} \Diamond (pc = \text{``Done''})
139
       END TRANSLATION
141
143
```