

global

Addr head; Addr tail;

$\boxed{Set\langle Addr \rangle reg; Set\langle Elem \rangle elems;}$

assume

$reg = \{head, tail, null\}$

$\wedge elems = \{head \rightarrow data, tail \rightarrow data\}$

$\wedge head \neq tail \wedge head \neq null \wedge tail \neq null$

$\wedge head \rightarrow data = -\infty \wedge tail \rightarrow data = +\infty$

$\wedge head \rightarrow next = tail \wedge tail \rightarrow next = null$

procedure MGC

Elem e

begin

1: **while** true **do**

2: *e* := **havocListElem**()

3: **nondet choice**

4: **call** SEARCH(*e*)

or

5: **call** INSERT(*e*)

or

6: **call** REMOVE(*e*)

7: **end while**

end procedure

procedure INSERT(*e*)

Addr prev

Addr curr

Addr aux

begin

23: *prev* := *head*

24: *prev* → *lock*()

25: *curr* := *prev* → *next*

26: *curr* → *lock*()

27: **while** *curr* → *data* < *e* **do**

28: *aux* := *prev*

29: *prev* := *curr*

30: *aux* → *unlock*()

31: *curr* := *curr* → *next*

32: *curr* → *lock*()

33: **end while**

34: **if** *curr* ≠ null ∧ *curr* → *data* > *e* **then**

35: *aux* := *malloc*(*e*, null, #)

36: *aux* → *next* := *curr*

37: *prev* → *next* := *aux*

$\boxed{reg := reg \cup \{aux\}}$

$\boxed{elems := elems \cup \{e\}}$

38: **end if**

39: *prev* → *unlock*()

40: *curr* → *unlock*()

41: **return**

end procedure

procedure SEARCH(*e*)

Addr prev

Addr curr

Addr aux

Bool found

begin

8: *prev* := *head*

9: *prev* → *lock*()

10: *curr* := *prev* → *next*

11: *curr* → *lock*

12: **while** *curr* → *data* < *e* **do**

13: *aux* := *prev*

14: *prev* := *curr*

15: *aux* → *unlock*()

16: *curr* := *curr* → *next*

17: *curr* → *lock*()

18: **end while**

19: *found* := (*curr* → *data* = *e*)

20: *prev* → *unlock*()

21: *curr* → *unlock*()

22: **return** *found*

end procedure

procedure REMOVE(*e*)

Addr prev

Addr curr

Addr aux

begin

42: *prev* := *head*

43: *prev* → *lock*()

44: *curr* := *prev* → *next*

45: *curr* → *lock*()

46: **while** *curr* → *data* < *e* **do**

47: *aux* := *prev*

48: *prev* := *curr*

49: *aux* → *unlock*()

50: *curr* := *curr* → *next*

51: *curr* → *lock*()

52: **end while**

53: **if** *curr* ≠ *tail* ∧ *curr* → *data* = *e* **then**

54: *aux* := *curr* → *next*

55: *prev* → *next* := *aux*

$\boxed{reg := reg \setminus \{curr\}}$

$\boxed{elems := elems \setminus \{e\}}$

56: **end if**

57: *prev* → *unlock*()

58: *curr* → *unlock*()

59: **return**

end procedure