Type specification for bset

bset = **type**

uses BSet (bset **for** *S*)
for all *s*: bset

invariant $max(s_p.elements) <= s_p.limit, min\ (s_p.elements) >= 0$.

% *max* is the value of the greatest element in *elements*.

% The invariant says the values in the set are in the range [0 … $s_p.limit$].

constraint $s_p.limit = s_q.limit$

insert = **proc** (*i*: int)

requires $i <= s_p.limit \wedge i >= 0$.

modifies *s*

ensures $s_{post}.limit = s_{pre}.limit \wedge s_{post}.elements = s_{pre}.elements \cup \{i\}$

contains = **proc** (*el*: int) returns (bool)

ensures result = $el \in s$

choose = **proc** () returns (int)

requires $s_{pre}.elements \neq \{\}$

modifies *s*

ensures $s_{post}.elements = s_{pre}.elements - \{result\}$

$\wedge\ result \in s_{pre}.elements \wedge s_{post}.limit = s_{pre}.limit$

size = **proc** () returns (int)

ensures $result = |\ s.elements\ |$

equal = **proc** (*t*: set) returns (bool)

ensures $result = (s = t)$

Is bset a subtype of bag (defined in Liskov and Wing Figure 1)?

Type specification for uset

uset = **type**

uses Set (uset **for** *S*)
for all *s*: uset

invariant true
constraint true

insert = **proc** (*i*: int)
    modifies *s*
    ensures $s_{post}.elements = s_{pre}.elements \cup \{i\}$
contains = **proc** (*el*: int) returns (bool)
    ensures result = $el \in s$

choose = **proc** () returns (int)
    requires $s_{pre}.elements \neq \{\}$
    modifies *s*
    ensures $s_{post}.elements = s_{pre}.elements - \{result\}$
        $\wedge\ result \in s_{pre}.elements \wedge s_{post}.limit = s_{pre}.limit$

size = **proc** () returns (int)
    ensures $result = |\ s.elements\ |$

equal = **proc** (*t*: set) returns (bool)
    ensures $result = (s = t)$

Is uset a subtype of bag?
Is uset a subtype of bset?
Is bset a subtype of uset?