

A Cecil predikátum objektumai

Az időben változó objektum viselkedés támogatása
forrás:

The Cecil Language Specification and Rationale

Version 3.2

Craig Chambers and the Cecil Group
Department of Computer Science and Engineering
University of Washington
Box 352350, Seattle, Washington 98195-2350 USA

A predikátum objektum szemantikája

- Mint normál objektum, de van egy hozzárendelt predikátum-kifejezése
- Ha egy objektum örököl az \bar{o} szüleitől, és a szülőhöz hozzá van rendelve egy olyan predikátum objektum is, aminek a predikátum-kifejezése igaz értéket ad, akkor minden ilyen predikátum objektumtól is örököl

Példa: buffer

```
object buffer isa collection;
  field elements(b@buffer); -- a queue of elements
  field max_size(b@buffer); -- an integer
  method length(b@buffer) { b.elements.length }
  method is_empty(b@buffer) { b.length = 0 }
  method is_full(b@buffer) { b.length = b.max_size }

predicate empty_buffer isa buffer when buffer.is_empty;
  method get(b@empty_buffer) { ... } -- raise error or block caller

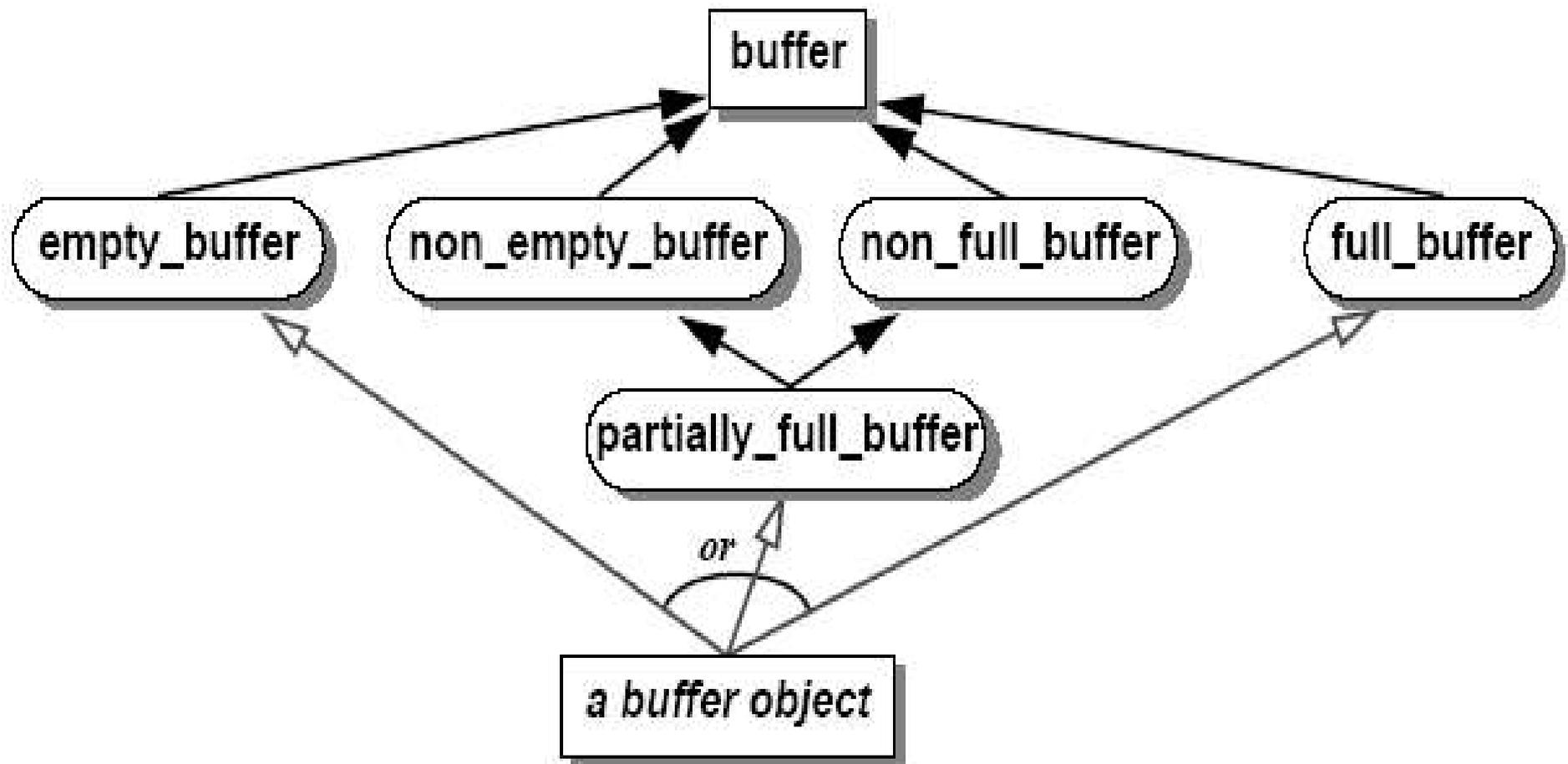
predicate non_empty_buffer isa buffer when not(buffer.is_empty);
  method get(b@non_empty_buffer) { remove_from_front(b.elements) }

predicate full_buffer isa buffer when buffer.is_full;
  method put(b@full_buffer, x) { ... } -- raise error or block caller

predicate non_full_buffer isa buffer when not(buffer.is_full);
  method put(b@non_full_buffer, x) { add_to_back(b.elements, x); }

predicate partially_full_buffer isa non_empty_buffer,
non_full_buffer;
```

Öröklődés kapcsolatok



(+ direkt öröklődés a buffer objektum, és buffer között)

Predikátum objektumok + öröklődés (1)

- Ha a következő két feltétel teljesül, akkor az O objektum a P predikátum objektum gyereke lesz
 - Az O a P minden egyes szülőjének leszármazottja, valamint
 - A P predikátum-kifejezése igaz, amikor kiértékeli, és ezt egy olyan hatókörön belül teszi, ahol a P bármely szülőjének neve hozzá köthető O -hoz
 - (automatikus)

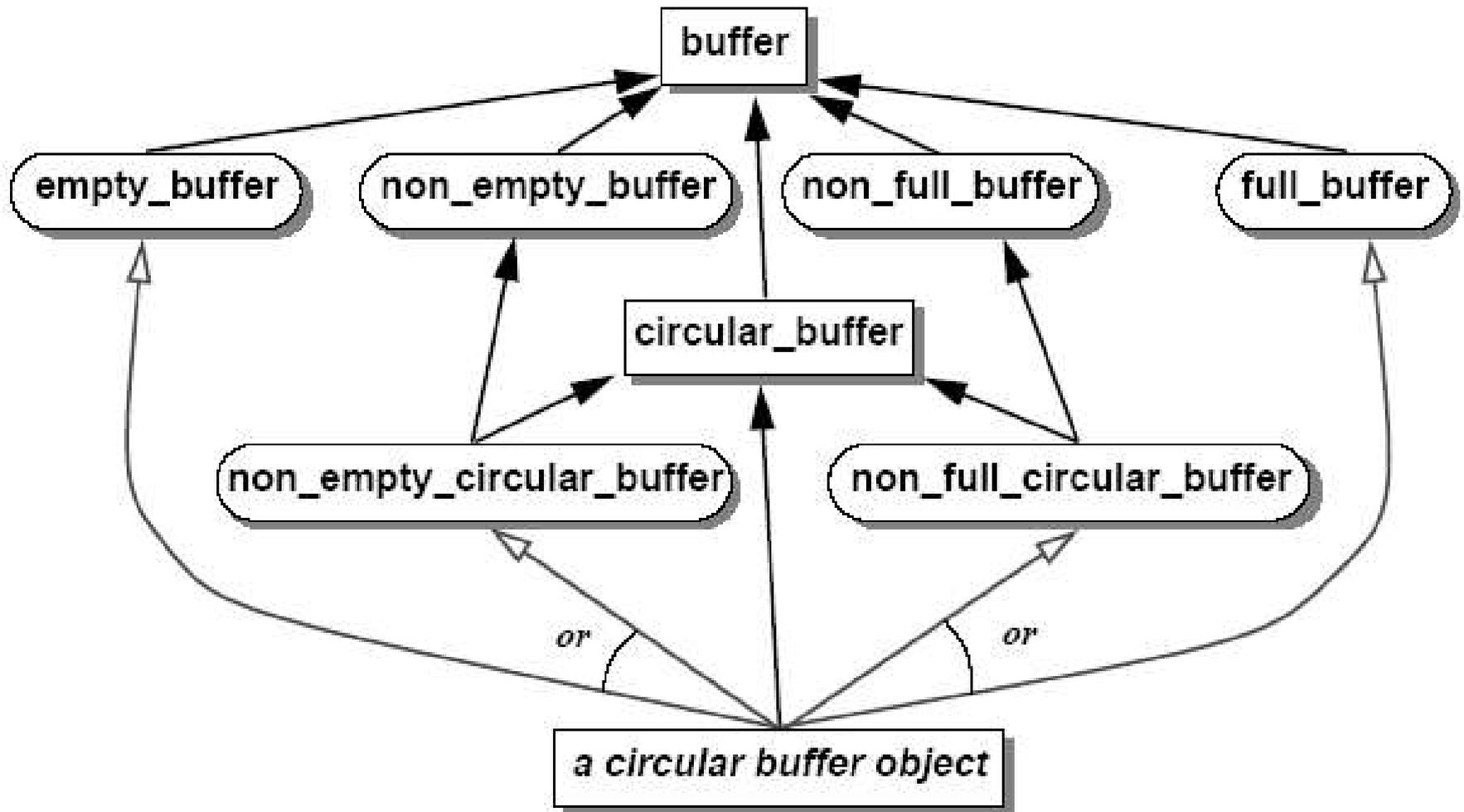
Predikátum objektumok + öröklődés (2)

- Ha egy predikátum egy másik predikátumtól örököl...
 - Ekkor az öröklő predikátum objektum predikátum-kifejezése csak abban az esetben értékelődik ki, ha a szülője igazként értékelődött ki...

Példa: circular_buffer

```
object circular_buffer isa buffer;
  field array(b@circular_buffer); -- a fixed-length array of elements
  var field insert_pos(b@circular_buffer); -- an index into the array
  var field remove_pos(b@circular_buffer); -- another integer index
  method max_size(b@circular_buffer) { b.array.length }
  method length(b@circular_buffer) {
    -- % is modulus operator
    (b.insert_pos - b.remove_pos) % b.array.length }
predicate non_empty_circular_buffer isa circular_buffer,
non_empty_buffer;
  method get(b@non_empty_circular_buffer) {
    var x := fetch(b.array, b.remove_pos);
    b.remove_pos := (b.remove_pos + 1) % b.array.length;
    x }
predicate non_full_circular_buffer isa circular_buffer, non_full_buffer;
  method put(b@non_full_circular_buffer, x) {
    store(b.array, b.insert_pos, x);
    b.insert_pos := (b.insert_pos + 1) % b.array.length; }
```

Öröklődés kapcsolatok



(+ `partially_full_buffer`)

Predikátum objektum attribútuma...

```
object window isa interactive_graphical_object;
var field iconified(@window) := false;
method display(w@window) {
    -- draw window using w.position
    ... }
method erase(w@window) {
    -- clear space where window is
    ... }
method move(w@window, new_position) {
    -- works for both expanded and iconified windows!
    w.erase; w.position := new_position; w.display; }
predicate expanded_window isa window when not(window.iconified);
var field position(@expanded_window) := upper_left;
field text(@expanded_window);
method iconify(w@expanded_window) {
    w.erase; w.iconified := true; w.display; }
predicate iconified_window isa window when window.iconified;
var field position(@iconified_window) := lower_right;
field icon(@iconified_window);
method open(w@iconified_window) {
    w.erase; w.iconified := false; w.display; }
```